Preliminary Information

This document contains information on a new product. The parametric information, although not fully characterized, is the result of testing initial devices.

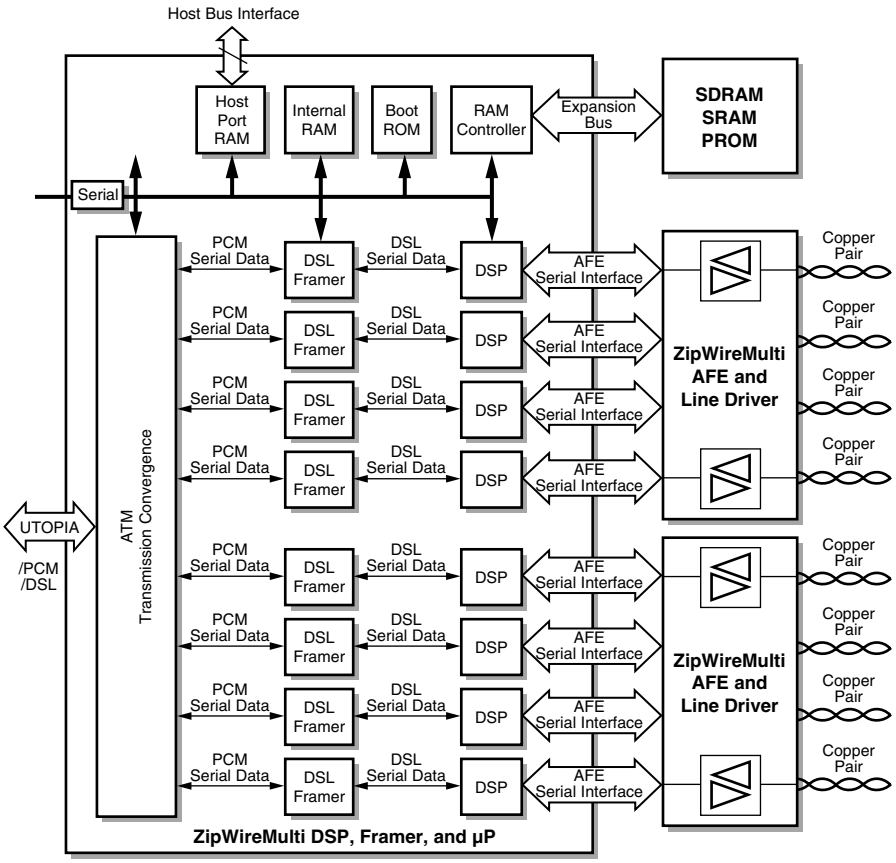
CX28985

Octal ZipWireMulti G.shdsl Transceiver with Embedded Microprocessor

Multi-Mode Operation: G.shdsl+, HDSL2, SDSL, HDSL, and IDSL

The ZipWireMulti DSL solution goes beyond simple compliance with the ITU G.shdsl standard by supporting the optional Enhanced Performance Asymmetrical PSD (EPAP) modes of operation. In addition, it is compliant with the ANSI HDSL2 standard (ANSI T1.418) and provides interoperability with Conexant's market-leading ZipWire transceivers through operation in 2B1Q multi-rate mode. The 2B1Q mode includes support of AutoBaud for SDSL interoperability, rate optimization, fast connect times, and standards-based HDSL operation. Furthermore, it operates in IDSL mode for interoperability with Basic Rate ISDN repeaters. The ZipWireMulti also supports Conexant's own proprietary modes, such as 32-PAM, 64 kbps, and 3.088 Mbps operation, which provide enhanced spectral compatibility, extended subscriber line reach, and high-speed operation. All these modes are supported by a single hardware circuit (i.e., one transformer, crystal, and hybrid for all modes) and can be configured in realtime via software control. (*Continued*)

Functional Block Diagram



Distinguishing Features

- Multimode operation including:
  - ITU G.shdsl including EPAP modes (ITU G.991.2)
  - ITU G.handshake (ITU G.994.1)
  - HDSL2 (ANSI T1.418)
  - SDSL/2B1Q (AutoBaud)
  - HDSL (ITU G.991.1, ETS 101 135 and ANSI TR-28)
  - IDSL (ANSI T1.601)
  - Proprietary/Extended Reach (ANSI spectrum management for loop transmission systems)
  - Proprietary/high-speed (ANSI spectrum management for loop transmission systems)
- Low power consumption of under 1 W at 2320 kbps, which includes the AFE and line driver dissipation
- Highly integrated solution including framer, microprocessor, ROM/RAM, frequency synthesizer, DSP, AFE, and line driver
- Embedded microprocessor for autonomous operation and EOC processing
- Data rates from 64 kbps to 3.088 Mbps in 8 kbps increments
- Interoperability with ZipWire 2B1Q transceivers including AutoBaud
- Simultaneous operation of UTOPIA Level 2 and PCM interfaces on a per-channel basis
- Central office (COT) and remote (RT) operation
- Individual clock recover circuits per channel
- Fast warm startup
- Glueless interface to popular microprocessors
- Single hardware circuit supports all speeds and modes of operation
- +1.8 V, +3.3 V and +12 V power supplies
- JTAG boundary scan

Ordering Information

Model Number	Package	Ambient Temperature Range
CX28985	27 × 27 mm PBGA 7 × 7 mm LGA	–40 °C to +85 °C

Revision History

Revision	Level	Date	Description
A	Advance	August 2000	Initial release of document number 101205C.
B	Advance	August 2000	Added “copyright information” to footer on Front page. No technical information was changed. Document number 101205C.
A	Preliminary	February 2001	Initial release under document number 500015A. Formerly document number 101205C.

Embedded Microprocessor

The ZipWireMulti chip set includes an embedded ARM microprocessor and a full suite of software that facilitate speedy, simplified development of systems compliant with all applicable ITU, ANSI, and ETSI standards. The embedded microprocessor and software handle the EOC processing and many other functions often delegated to an external host controller in competing solutions, greatly reducing software porting efforts and eliminating real-time processing requirements for an external host controller. The host controls the ZipWireMulti through a simple and well-defined software API common to all devices in the Conexant ZipWire SDSL and AccessRunner ADSL family.

Flexible Framer Supports Simultaneous Operation of the UTOPIA and PCM Interfaces

The ZipWireMulti integrated framers include fully featured UTOPIA Level 2 and PCM interfaces. The UTOPIA interface includes ATM TC layer processing. The DSL payload can be mapped to either interface, or split between them to enable simultaneous connections to both ATM- and TDM-based systems. This feature is programmable on a per-port basis, and supports a mix of channels transporting either data, data plus time slot voice, or only time slot voice. The framer automatically extracts and inserts DSL overhead (i.e., EOC, indicator and Z-bits, CRC, sync word, etc.) and passes it to the embedded microprocessor for processing. In addition, the framer supports all G.shdsl, HDSL2, HDSL, and IDSL frame formats, EOC messaging protocols, and other non-standard SDSL frame formats. For non-framed protocols, the ZipWireMulti supports a framer bypass mode.

Full System Solution Includes Integrated Line Driver and Frequency Synthesizer

The ZipWireMulti includes an integrated line driver and frequency synthesizer to provide a full DSL solution. The integrated line driver is capable of driving the high line power EPAPs for payload rates of 768, 1544, 2048, and 2304 kbps according to the G.shdsl standard. The frequency synthesizer, along with the rest of the ZipWireMulti, supports data rates from 64 kbps to 3.088 Mbps, and requires only one external crystal. This highly integrated DSL solution enables OEMs to design and manufacture the most feature-rich, lowest power, and highest density DSL equipment in the industry.

Applications

- Digital Subscriber Line Access Muxes (DSLAMs)
- DSL-enabled Digital Loop Carriers (DLC)
- Nx64K data transport
- Remote LAN access
- T1 and E1 HDSL-enabled transport systems
- Cellular Base Station data links
- Campus modems
- Data, voice, and video transport systems

Table of Contents

List of Figures

List of Tables

1.0 System Overview

1.1 Introduction

This data sheet provides information for using the CX28985 ZipWireMulti G.shdsl Transceiver and Framer, and ZipWireMulti Analog FrontEnd (AFE) devices. It describes application and hardware interfaces. In addition, it provides detailed descriptions of the devices and pins, configuration information, and electrical and mechanical specifications.

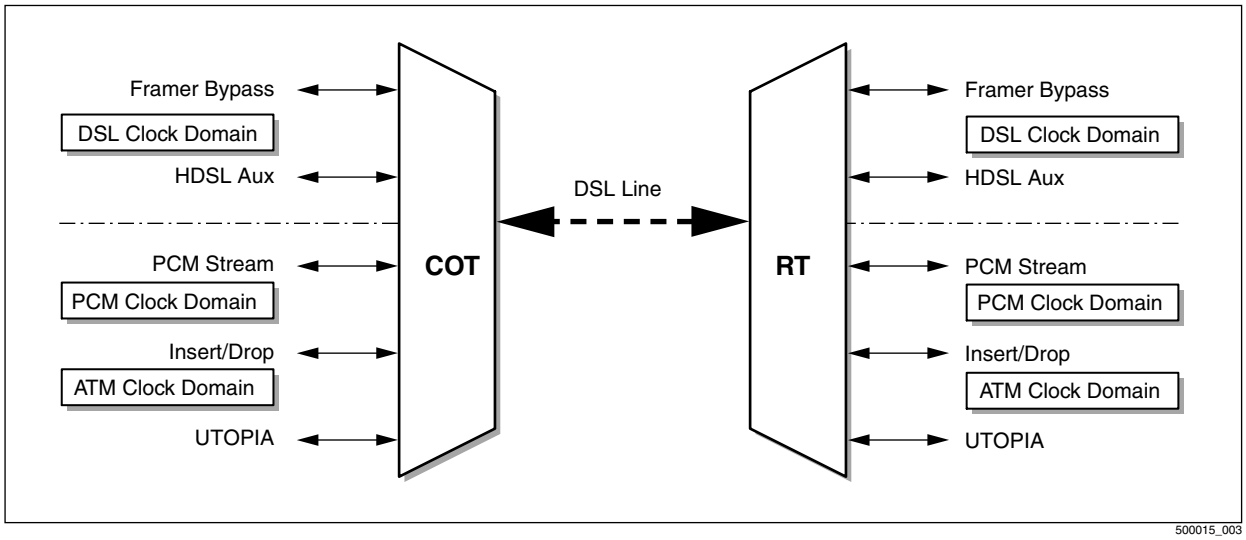
Because of the ZipWireMulti chip set's flexibility, not all applications are addressed in this data sheet. Please contact the local sales office or technical support to determine how the ZipWireMulti device can be used in your DSL application.

1.2 References

T1/E1.4 (T1E1.4/99-006)—Draft for HDSL2 Standard
RE/TM-06011-1—Draft for SDSL

For most applications, the ZipWireMulti chip set can be viewed as a pair of wires: what comes in on one terminal unit goes out the far-end terminal unit. The Framer Bypass and DSL auxiliary interfaces operate at the DSL line rate. The PCM and Insert/Drop operate at the PCM clock rate. The DSL line interfaces to the physical twisted pairs. Figure 1-1 illustrates the ZipWireMulti data interfaces.

Figure 1-1. High-Level Functional Diagram

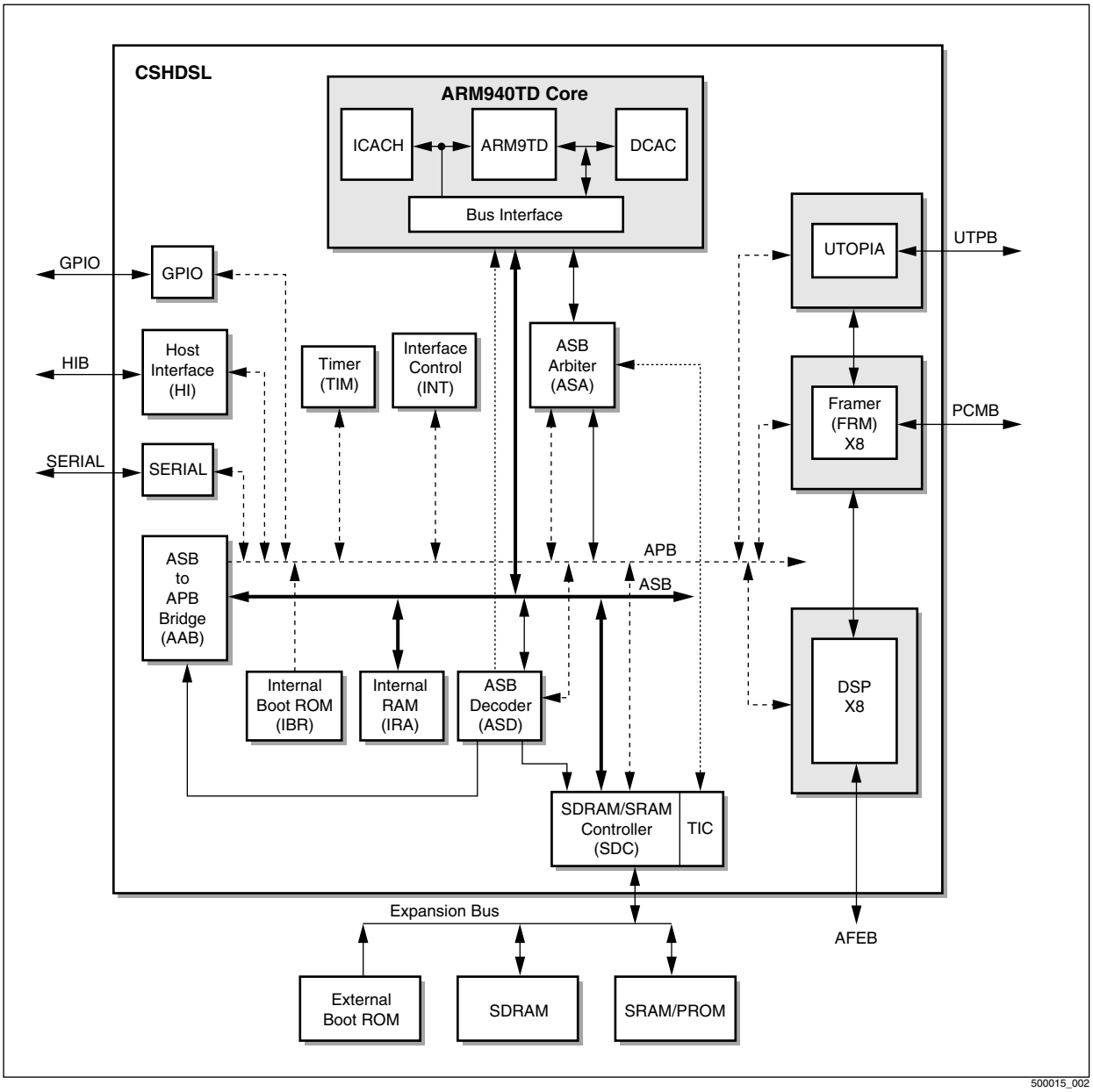


1.3 ZipWireMulti Transceiver/Framer Functional Summary

A full-featured API command set allows the user to configure the ZipWireMulti system, query for status, execute loopbacks and test modes, and dictate the program flow.

Figure 1-2 illustrates a detailed block diagram of the ZipWireMulti Transceiver/Framer.

Figure 1-2. ZipWireMulti Transceiver/Framer Detailed Block Diagram



1.3.1 The ZipWireMulti ARM Functional Summary

The ZipWireMulti Transceiver/Framer has a built-in ARM microprocessor core with the following features:

- Internal 2 KB non-programmable (masked) program ROM
- Host Interface
- One Asynchronous Serial (RS232) Interfaces
- Miscellaneous internal memory mapped peripherals
- Programmable chip select decoder
- Internal timers/counters

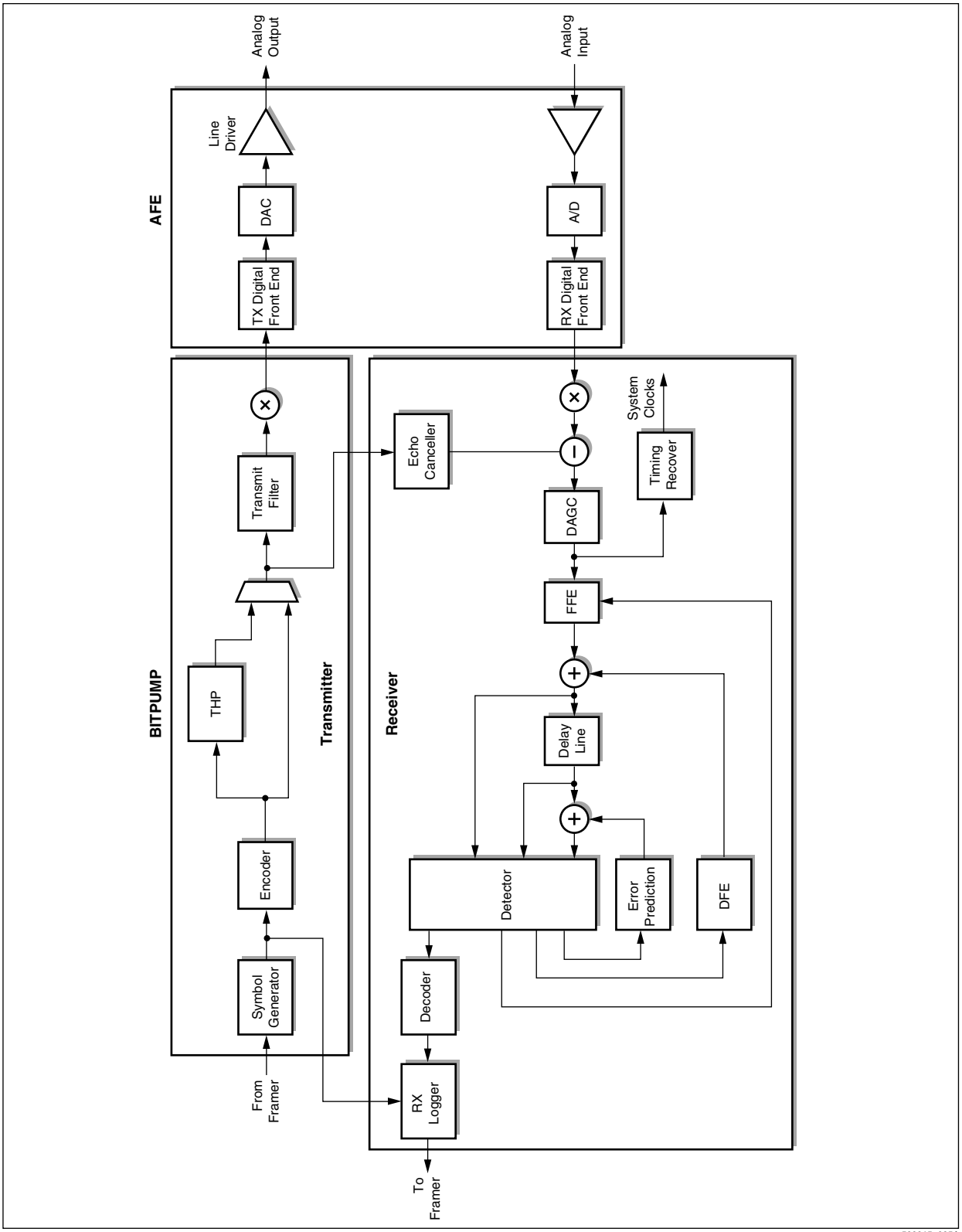
1.3.2 ZipWireMulti Transceiver/DSP Functional Summary

The transmitter receives a bit stream from the DSL Framer and maps the data bits to the appropriate PAM symbols. An optional precoding block supports both Tomlinson-Harashima precoding. The signal is then processed by the transmit filter to achieve the desired time and/or frequency domain characteristics before being forwarded to the Analog Front-End (AFE).

The receiver receives serialized data from the AFE device and from precoded symbols from the bit pump transmitter. The precoded symbols feed into an Echo Canceler (EC) which estimates the echo response and subtracts it from the AFE samples. The signal is equalized using a Feed Forward Equalizer (FFE) and a Decision Feedback Equalizer (DFE). Finally, a Trellis-Coded Modulation (TCM) decoder recovers the information bits. The DFE is used only during startup. An error predictor is used as a part of the startup algorithm and as a precoder coefficient adaptation machine during normal operation.

Figure 1-3 illustrates a detailed block diagram of the ZipWireMulti Transceiver/DSP section.

Figure 1-3. ZipWireMulti Transceiver/DSP Detailed Block Diagram

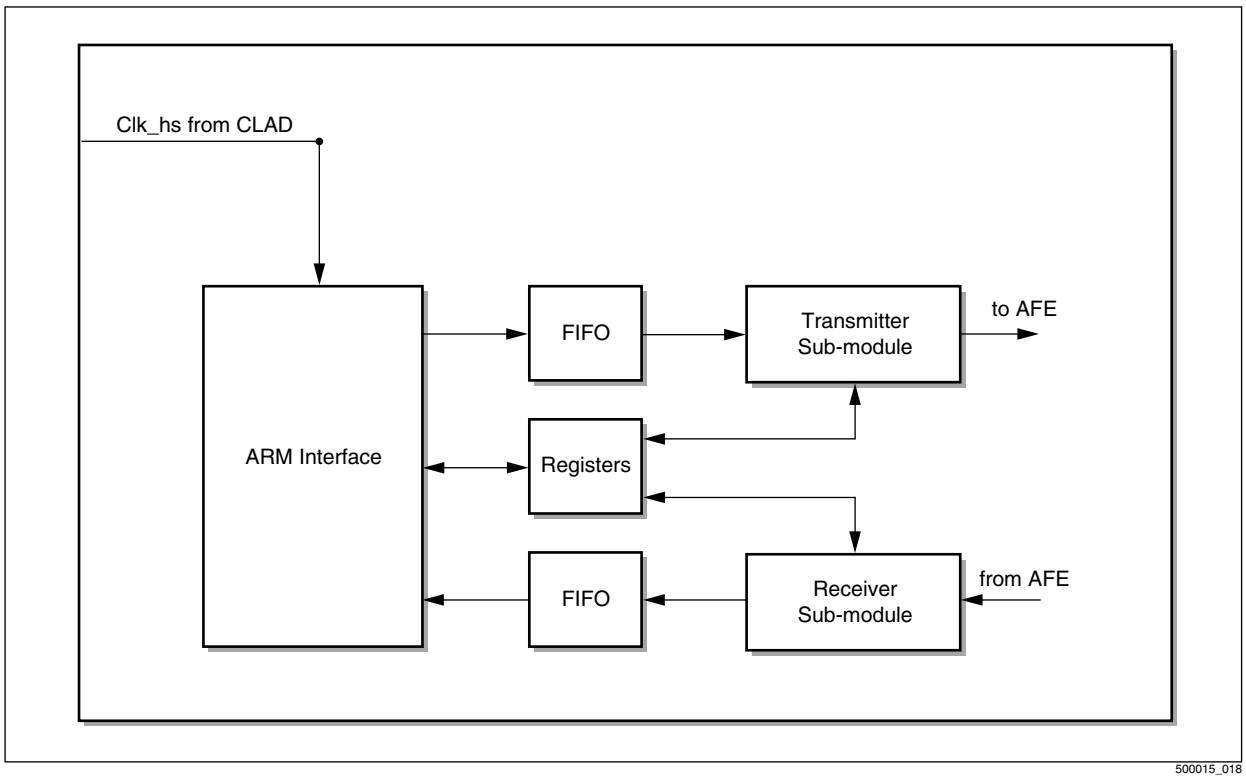


1.4 ZipWireMulti G.handshake Functional Summary

The G.handshake block implements the G.994.1 handshaking function. The G.994.1 standard defines the signals, messages, and procedures for exchanging information regarding the capabilities of each transceiver and for selecting common modes of operation. This block implements a DPSK modem that operates at 800 symbols/second. This block has interfaces to the CLAD, ARM microcontroller, the AFE, and the PLL block.

Figure 1-4 illustrates a detailed block diagram of the ZipWireMulti G.handshake section.

Figure 1-4. G.handshake Block Diagram

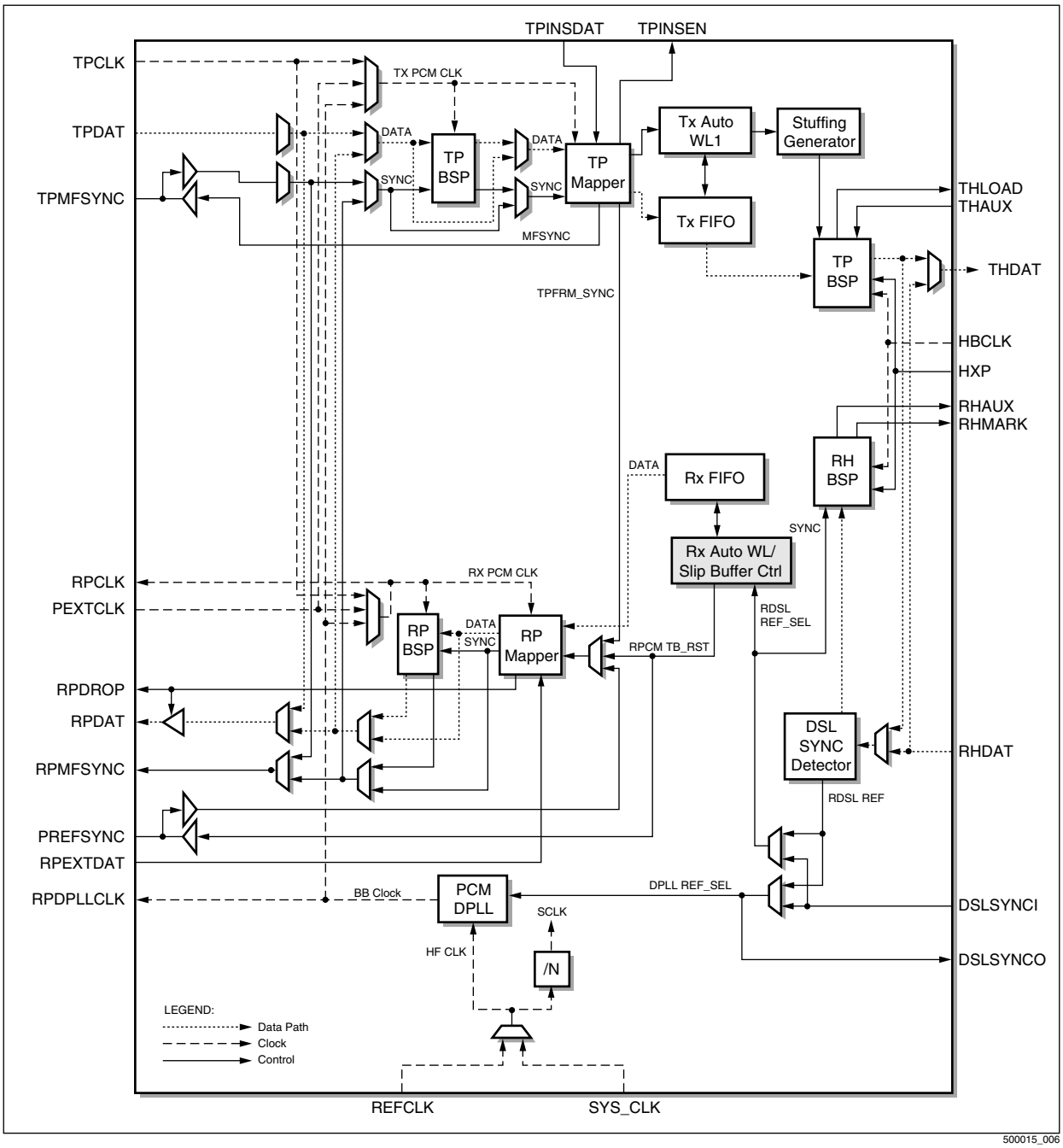


1.5 ZipWireMulti DSL Framer Functional Summary

The DSL Framer supports HDSL1, HDSL2, G.shdsl, and custom frame structure applications. The DSL Framer provides clock, data, and frame format conversion from various PCM frame formats to various DSL applications. The DSL Framer supports Multipair configuration such as T1 two loops, E1 two, or any Point-to-Multipoint (P2MP) application by cascading several DSL Framers. The DSL Framer provides full PCM termination capabilities, including synchronization and management of E1 PRA and T1. The DSL rate can vary from 144 kbps up to 3.088 Mbps and the PCM rate can vary from 64 kbps up to 16.384 Mbps (8 × E1) and any custom PCM rate and frame format within this range. Chapter 4.0 describes the details of the DSL Framer section.

Figure 4-1 illustrates a detailed block diagram of the ZipWireMulti DSL Framer section.

Figure 1-5. DSL Framer Detailed Block Diagram



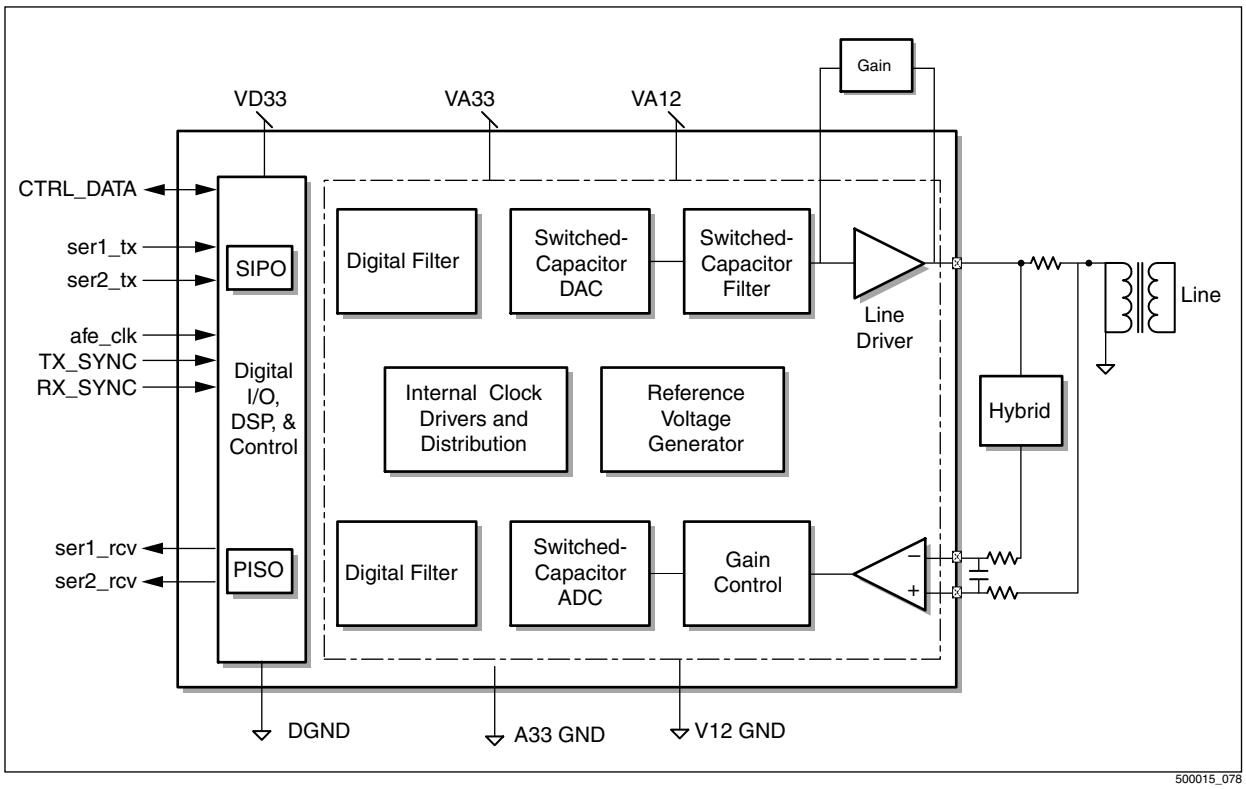
1.6 ZipWireMulti AFE Functional Summary

The ZipWireMulti AFE performs the analog functions for transmission and reception of G.shdsl, HDSL2 OPTIS or HDSL1 2B1Q line-code signals. ZipWireMulti AFE includes the Digital-to-Analog (D/A) and Analog-to-Digital (A/D) conversion, data converter anti-aliasing and post filtering, gain control, and line driving.

The ZipWireMulti AFE serial digital interface connects to the ZipWireMulti Transceiver/Framer device. The serial interface protocol is proprietary. The DSP transceiver indirectly controls the AFE. The analog interface consists of the line driver feedback resistors, impedance matching resistors, external hybrid, and transformer.

Figure 1-6 illustrates a detailed block diagram of the ZipWireMulti AFE.

Figure 1-6. ZipWireMulti AFE Block Diagram



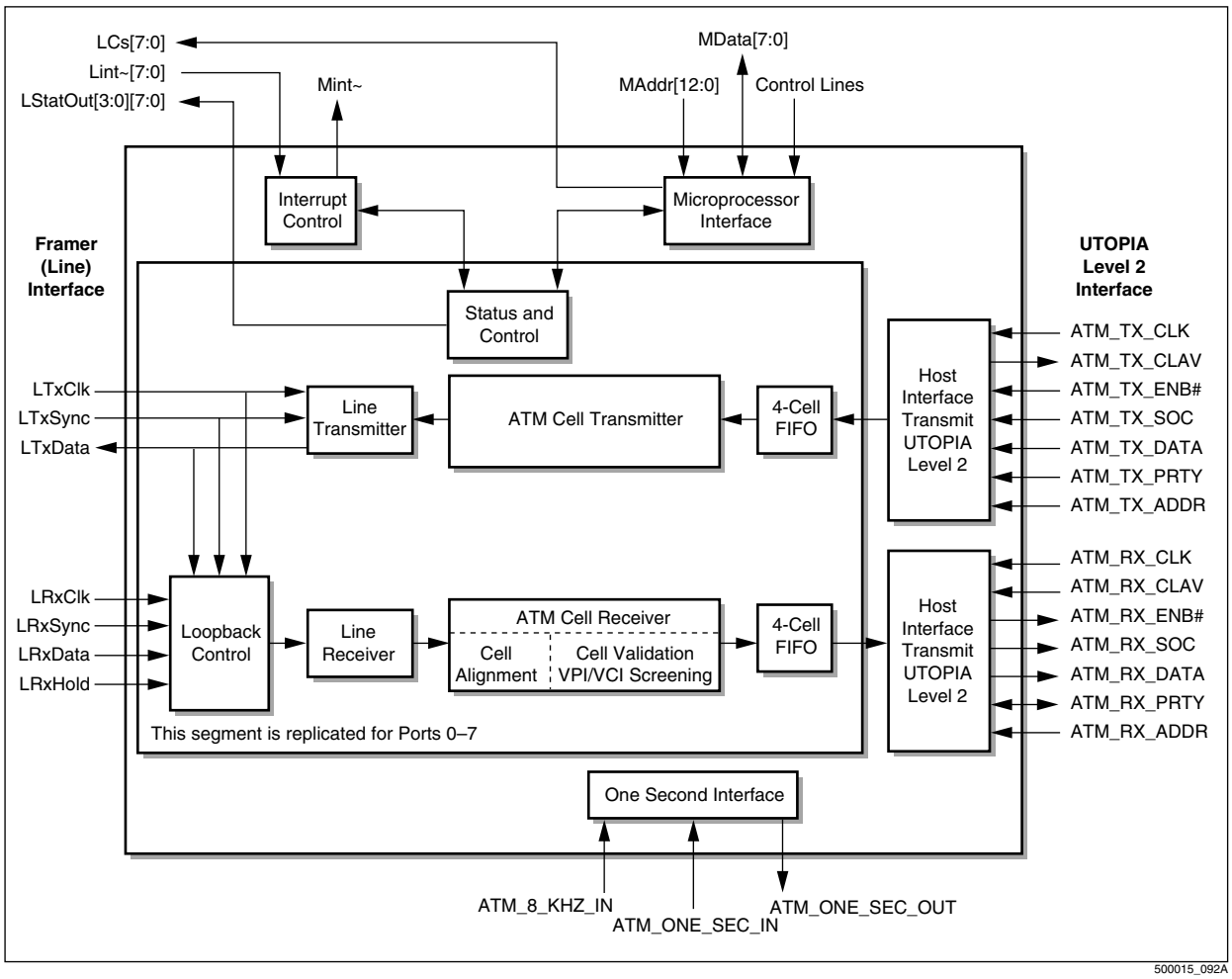
1.7 ZipWireMulti ATM Phy Transmission Convergence Functional Summary

The ATM Phy supports data rates ranging from 64 kbps to 50 Mbps. A UTOPIA Level 2 multi-Phy interface connects the ZipWireMulti to the host switch or terminal system. The ATM Phy performs all cell alignment functions on the bit stream. This gives system designers a simple, modular, and low-cost architect for supporting all UNI and NNI ATM interfaces. The details of the ATM Phy are described in Chapter 6.0.

NOTE: The ATM Phy TC block is a single slice of the RS8228 Octal TC Phy device.

Figure 1-7 illustrates a detailed block diagram of the ZipWireMulti ATM Physical Layer (Phy) Transmission Convergence (TC) section.

Figure 1-7. UTOPIA Phy TC Functional Block Diagram



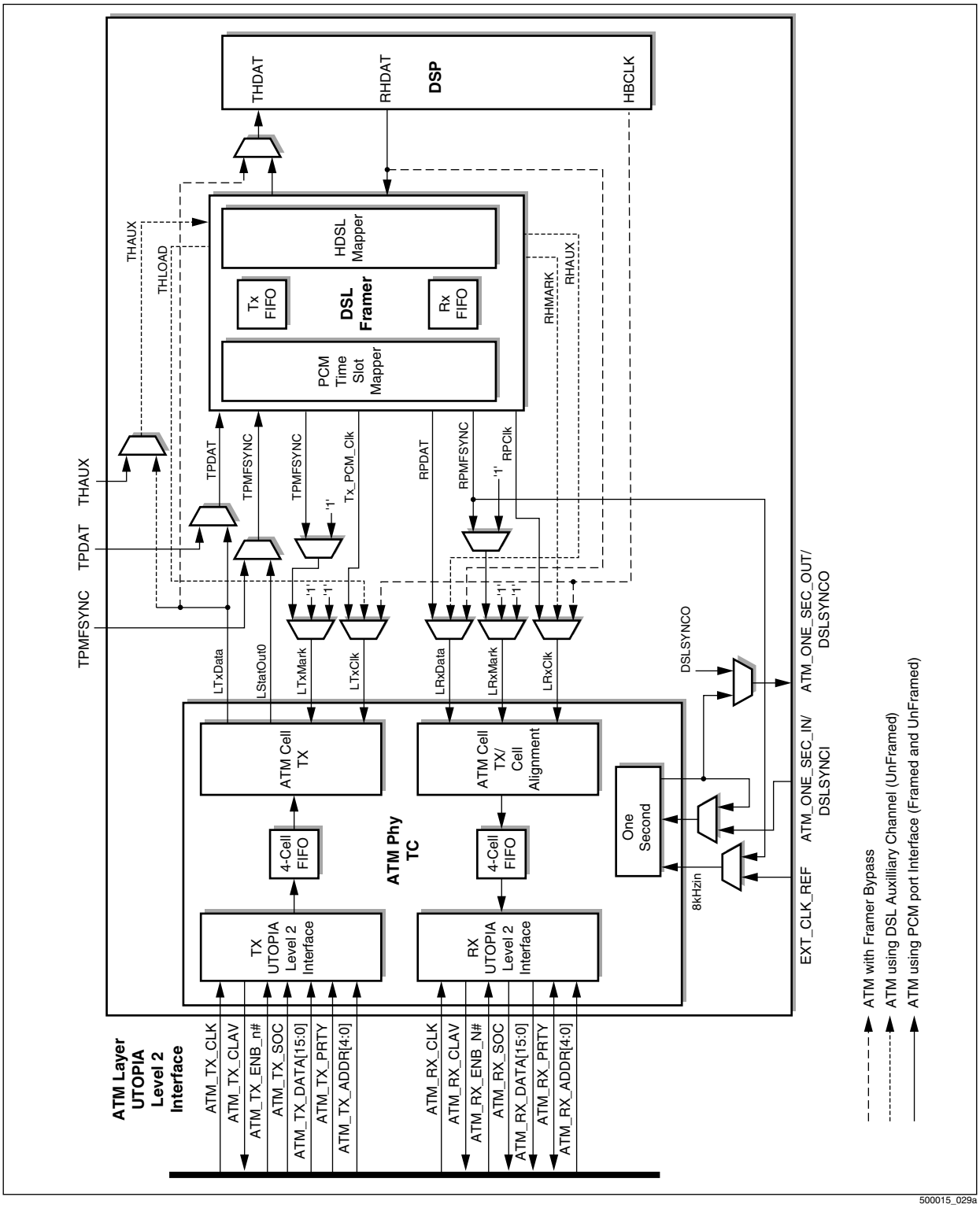
1.8 ZipWireMulti General Multiplexing Functional Summary

The general multiplexing block contains all muxes used to select different functions of the device by selecting different internal connections and different I/O configurations. The general multiplexing block details for each mode are described in Chapter 6.0. The following modes are supported:

- UTOPIA with DSL Framer enabled (ATM applications)
- UTOPIA using DSL Framer auxiliary channel (ATM applications)
- UTOPIA with DSL Framer bypassed (legacy Bt8973 with ATM)
- Serial ATM interface with DSL Framer enabled or bypassed (ATM applications)
- PCM interface using DSL Framer enabled (E1/T1 transport)
- PCM interface using DSP interface (legacy Bt8973)
- DSL Framer auxiliary channel (non standard applications)

Figure 1-10 illustrates a block diagram of the general multiplexing section.

Figure 1-8. General Multiplexing Functional Block Diagram

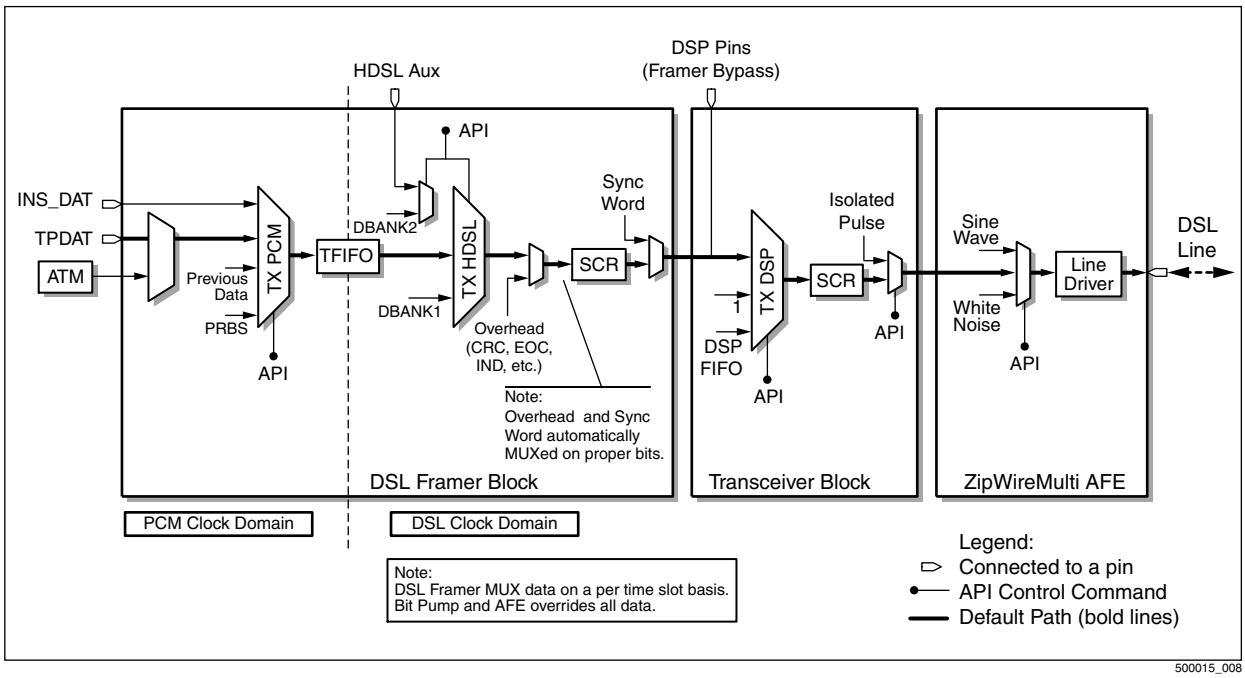


1.9 ZipWireMulti Transmit Path

Figure 1-9 illustrates the various input data sources sent out of the ZipWireMulti link. This drawing includes all external inputs and internally generated data sources, but does not illustrate loopbacks.

NOTE: Figure 1-9 does not show the ATM data path.

Figure 1-9. Detailed Transmit Data Path Block Diagram

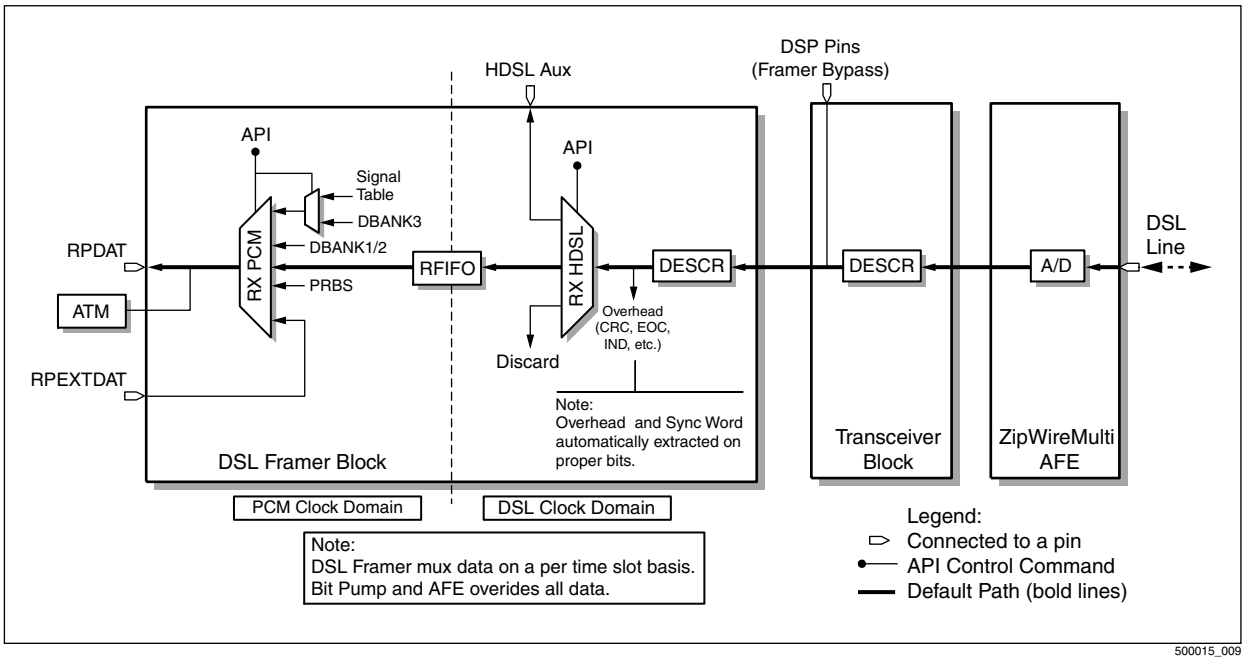


1.10 ZipWireMulti Receive Path

Figure 1-10 illustrates the various output destinations received from the ZipWireMulti link. This drawing includes all external inputs, and internally generated data sources, but figure does not illustrate loopbacks.

NOTE: Figure 1-10 does not show the ATM data path.

Figure 1-10. Detailed Receive Data Path Block Diagram



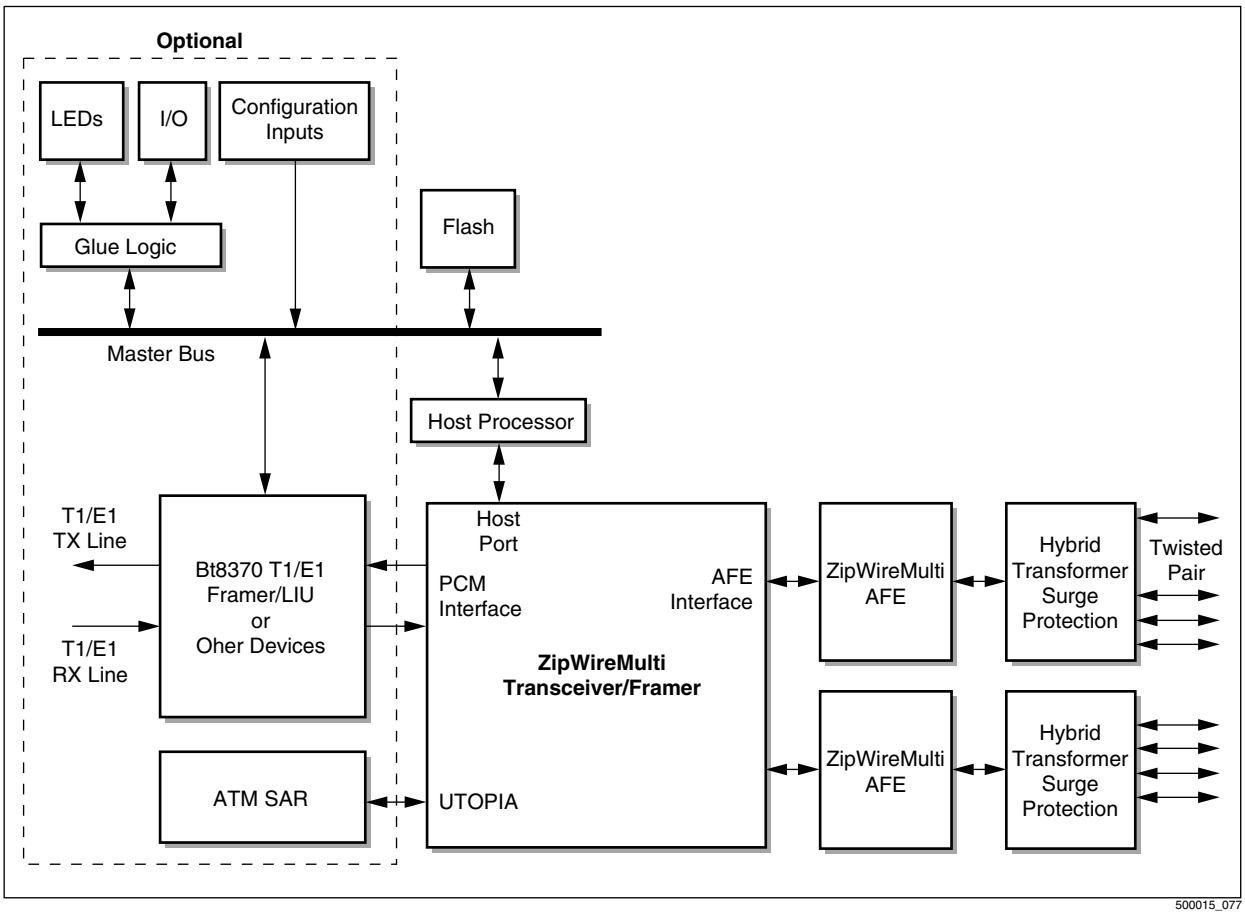
2.0 Application Interfaces

This section illustrates various application configurations. Each figure illustrates a different interface configuration.

2.1 Eight Port Configurations

Figures 2-1 and 2-2 illustrate the block diagrams for a single device, 8-port ZipWireMulti system.

Figure 2-1. Eight Port Hardware Configuration



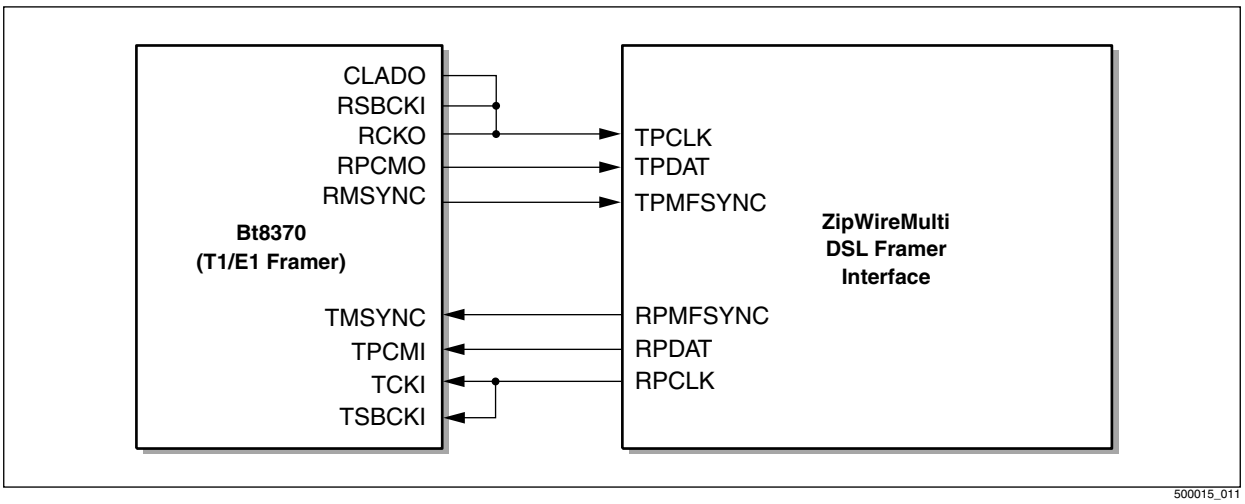
2.2 Framer Transparent Mode

Framers transparent mode would be required when the system needs to support interoperability with legacy HDSL1 applications that do not use the DSL Framer as well as HDSL2 applications that use the DSL Framer. When the DSL framer is configured to transparent mode, the data passes through the framer, without adding any overhead. To configure the DSL Framer for Transparent mode, the Frame Structure parameter in DSL_SYSTEM_CONFFIG API (Opcode 0x06) should be set to _TRANSPARENT_FORMAT (value 0x05).

2.3 ZipWireMulti Transceiver/Framer to Bt8370 T1/E1 Interface

Figure 2-2 illustrates one possible configuration for the ZipWireMulti Transceiver/Framer to Bt8370 T1/E1 Interface.

Figure 2-2. ZipWireMulti Transceiver/Framer to Bt8370 T1/E1 Interface

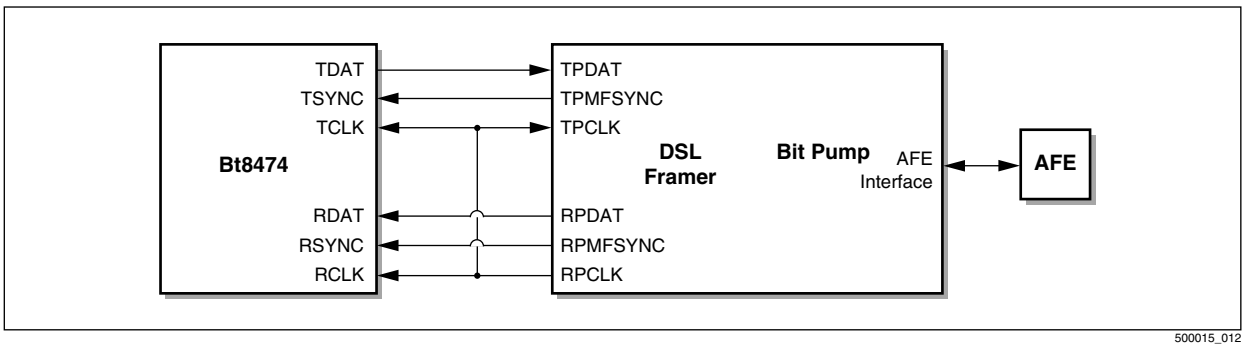


2.4 DSL Framer to Bt8474 Interface

Figure 2-3 illustrates how to connect the ZipWireMulti device to the Bt8474 device when using the ZipWireMulti DSL Framer block. In an HTU-C (central office) application, the DSL Framer DPLL is programmed to open loop mode to provide the clock reference. In an HTU-R (remote terminal) application, the DSL Framer DPLL is programmed to closed loop mode to recover the PCM clock reference from the HTU-C. The DSL Framer generates the transmit and receive multiframe sync reference and feeds it to the Bt8474 device. The multiframe sync signals would only be required in channelized applications where individual time slots are sourced from different devices.

Figure 2-3 illustrates only one port connection.

Figure 2-3. DSL Framer to Bt8474 Interface Diagram



2.5 Deliverables

The previous pages in this chapter provided an overview of several ZipWireMulti applications. The following provides a list of the deliverables provided along with the chip set.

- Data Sheet
- Hybrid Component Values
- Object File—for customers to download from the host processor
- Host Processor Application Code Examples

3.0 ZipWireMulti DSP Detailed Description

3.1 Clocking Architecture Overview Description

The CX28985 provides a flexible clocking architect to support the multiple number of applications defined by the various standards (G.shdsl, SDSL, HDSL1, HDSL2, and etc.). Table 3-1 lists the supported clocking modes. In general, the HTU-C clocking architect must support the different modes while the HTU-R clock reference is derived from the received symbol clock. The HTU-C specifics are described in Section 3.2 while the HTU-R specifics are described in Section 3.3.

Table 3-1. Clocking Modes

Mode #	Mode Name	HTU-C Clock Reference	Application
1	Plesiosynchronous	Local oscillator (free running)	Classic HDSL
2	Plesiosynchronous with timing reference	Network reference clock	Classic HDSL
3	Synchronous	Transmit data clock or network reference clock	—
4	Hybrid	Transmit data clock	Downstream is synchronous while upstream is Plesiosynchronous.

3.1.1 CX28985 Clocking Architect Implementation

The on-chip clock synthesizer (PLL) block is responsible for generating the DSP, Microprocessor, AFE, DSL Framer, and ATM reference clocks. The ZipWireMulti DSL Framer has a DPLL to generate the PCM received clock.

The clock synthesizer can either operate in free run, network timing reference, or HTU-R timing recovery mode. In free run mode, the ZipWireMulti device operates at the crystal (or local clock) phase offset. In network timing reference mode, the ZipWireMulti phase-locks its timing to the external reference clock. In HTU-R timing recovery mode, the ZipWireMulti phase-locks its timing to the DSL line (HTU-C).

Figure 3-1 illustrates the CX28985 clock tree distribution.

Figure 3-1. CX28985 Clock Tree Distribution

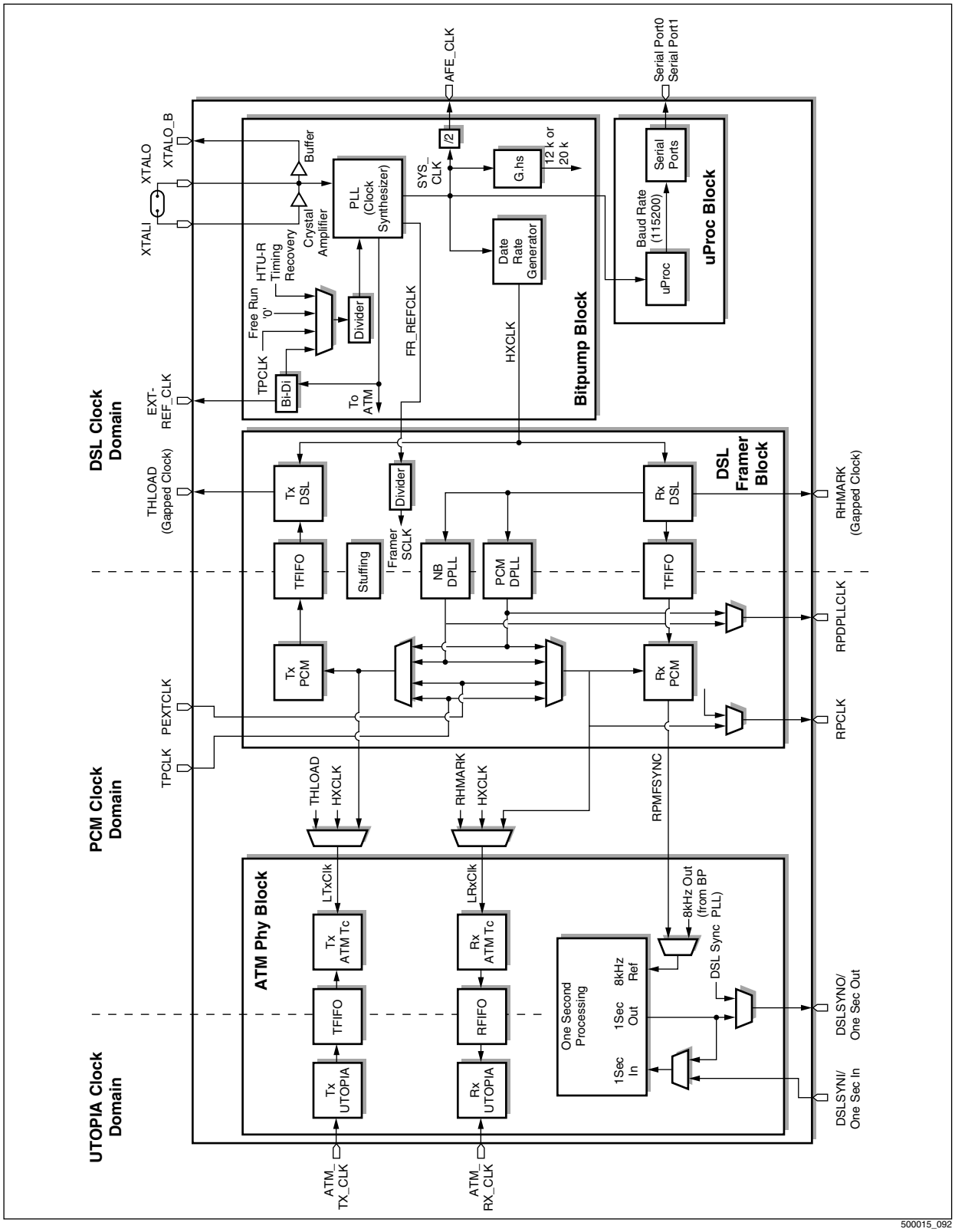


Table 3-2. CX28985 Clocks

Clock	Frequency	Description
Crystal	16.384 MHz	External Crystal or Clock Input
XTALI/XTALO	16.384 MHz	Crystal Input/Output
XTALO_B	16.384 MHz	Buffered Crystal Output
EXT_CLK_REF	8 kHz	Bidirectional network timing reference clock
8 kHz In	8 kHz	Internal 8 kHz input derived from EXT_CLK_REF or TPCLK
8 kHz Out	8 kHz	Internal 8 kHz output from PLL. Derived from 8k Hz In when HTU-C or recovered from DSL line when HTU-R
SYS_CLK	43–53 MHz	Internal DSP system clock
AFE_CLK	SYS_CLK / 2	AFE Clock Reference
G.hs Clock	12 kHz or 20 kHz	G.hs clock —12 kHz (HTU-R) or 20 kHz (HTU-C)
Baud Rate	115200	Baud rate that controls serial ports 0 and 1
HXCLK	64 k–3088 kbps	Data Rate Clock Output
FR_REF_CLK	~160 MHz	Internal DSL Framer reference clock
Framer SCLK	~15–20 MHz	Internal DSL Framer system clock

3.1.2 Network Reference Clock Input

Either be from external pin or from transmit PCM clock (TPCLK).

3.2 HTU-C Clocking Modes

This section describes how the various clocking modes are targeted for HTU-C applications.

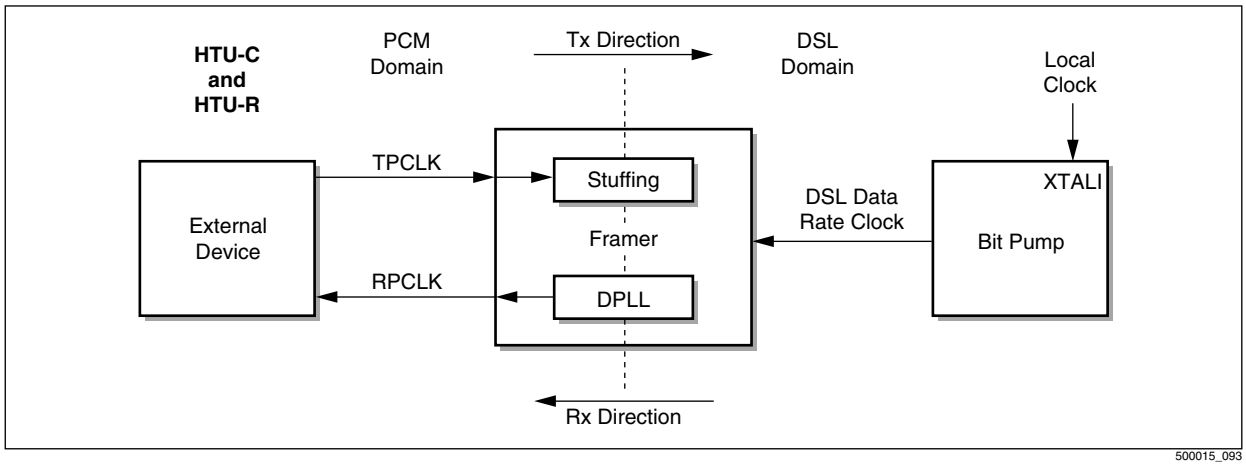
3.2.1 Plesiosynchronous Mode

In this mode, the stuffing generator compensates for any phase differences between the DSL and PCM clock domains. The transmit and receive PCM clocks can operate at independent rates (within the appropriate PPM tolerance). T1/E1 Transport applications use this mode.

NOTE: Figure 3-2 applies to both the HTU-C and HTU-R.

Figure 3-2 illustrates a simplified block diagram of the plesiosynchronous clock mode.

Figure 3-2. Plesiosynchronous Mode Block Diagram



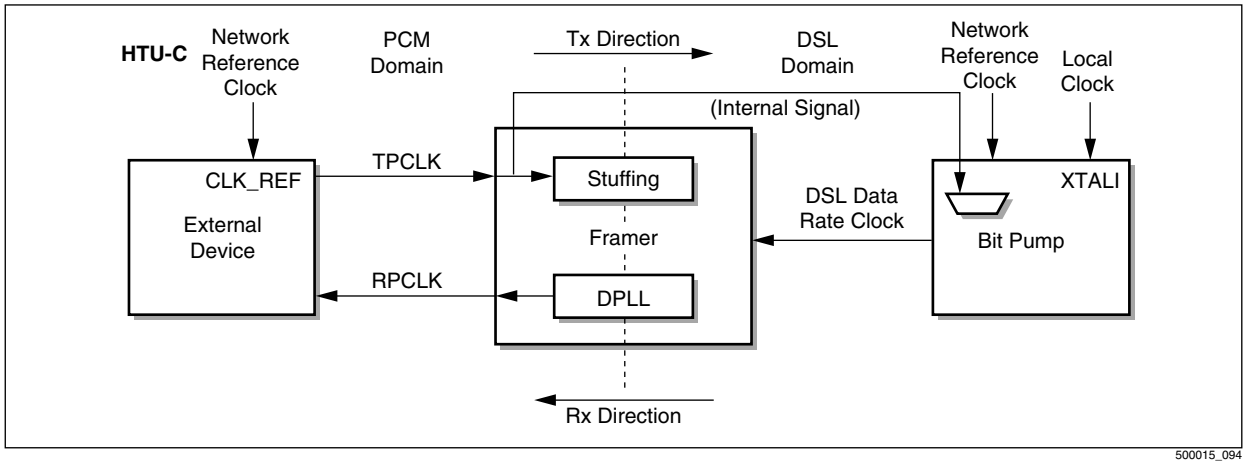
3.2.2 Plesiosynchronous Mode with Network Reference Clock

This mode is similar to the plesioschronous mode except the HTU-C locks the DSL clock to a network reference clock, either from an external network reference clock or the transmit PCM clock.

If the network reference clock is sourced from the TPCLK pin, then the DSL and PCM clock domains are synchronized. However, the stuffing generating is still used and is therefore a slightly different configuration than the synchronous modes (Section 3.2.3). This configuration is when a customer already has a networked timed PCM clock and does not wish to redundantly supply the network reference clock to the external clock reference pin.

Figure 3-3 illustrates a simplified block diagram of the plesiosynchronous mode with a network reference timing clock.

Figure 3-3. Plesiosynchronous Mode with Network Reference Clock Block Diagram



3.2.3 Synchronous Mode

In synchronous mode, the DSL and PCM clock domains are synchronized, and therefore, the stuffing generator is disabled. The synchronous mode can be achieved in three different ways:

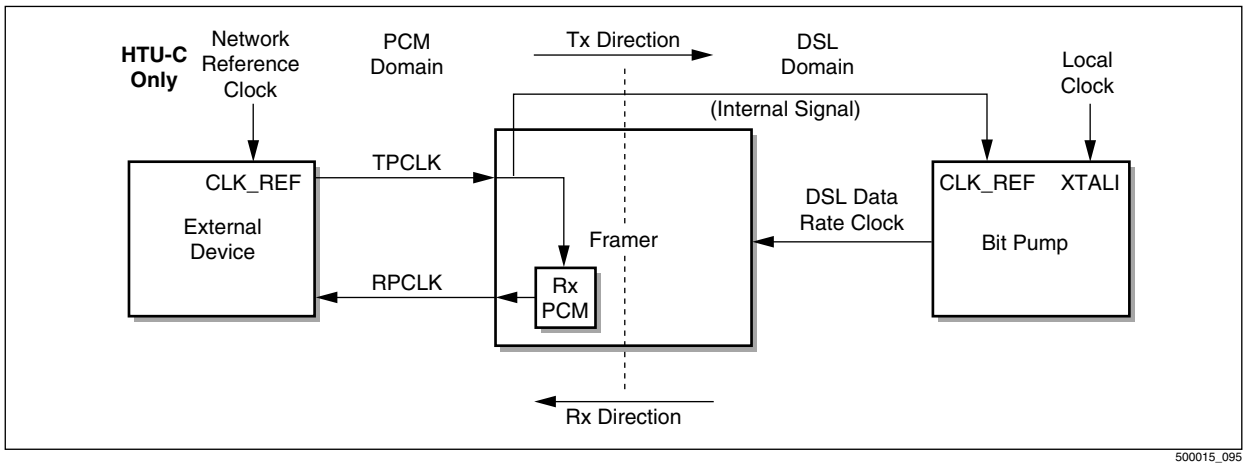
1. Synchronous—Slave Transmit Data Clock
2. Synchronous—Master Transmit Data Clock
3. Synchronous—Network Reference Clock

3.2.3.1 Synchronous Transmit Data Clock Slave PCM

The DSL clock domain synchronizes the local oscillator to the transmit PCM clock (TPCLK). The transmit PCM clock (TPCLK) and receive PCM clock (RPCLK) are both sourced (slaved) from the TPCLK input pin. The TPCLK must be supplied from an external source. The DPLL is disabled.

Figure 3-4 illustrates a simplified block diagram of the transmit data clock mode.

Figure 3-4. Synchronous Mode with Transmit Data Clock Block Diagram



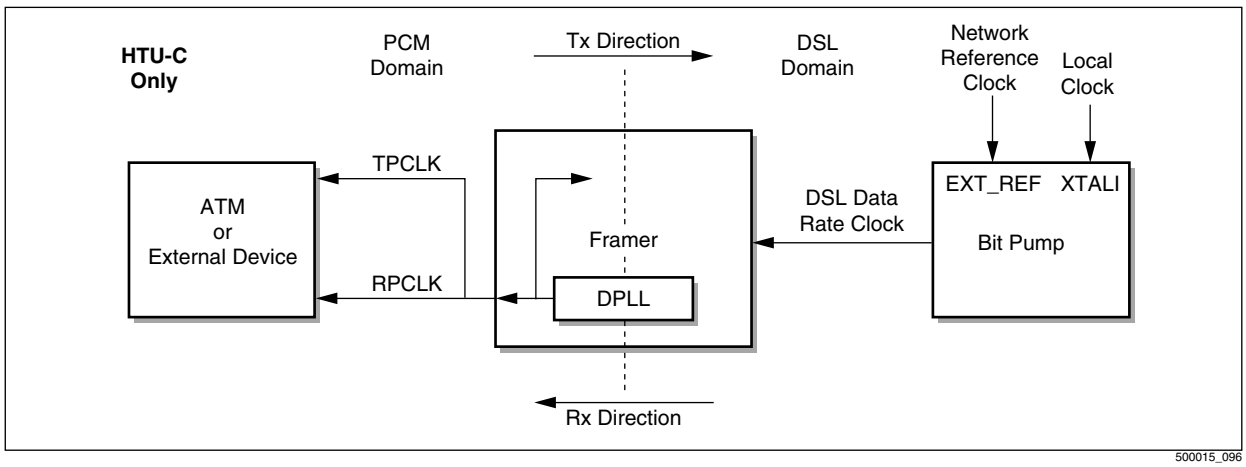
3.2.3.2 Synchronous Network Reference Clock Master PCM

In this configuration, the CX28985 is the PCM clock master. Any external device must be slaved to the CX28985 clocks. The DSL clock domain synchronizes the local oscillator to the external network reference clock. The transmit PCM clock (TPCLK) and receive PCM clock (RPCLK) then synchronize to the DSL clock using the DPLL.

NOTE: The RADSL application uses this clocking scheme.

Figure 3-5 illustrates a simplified block diagram of the master transmit data clock mode.

Figure 3-5. Synchronous Mode with Master Transmit Data Clock Block Diagram

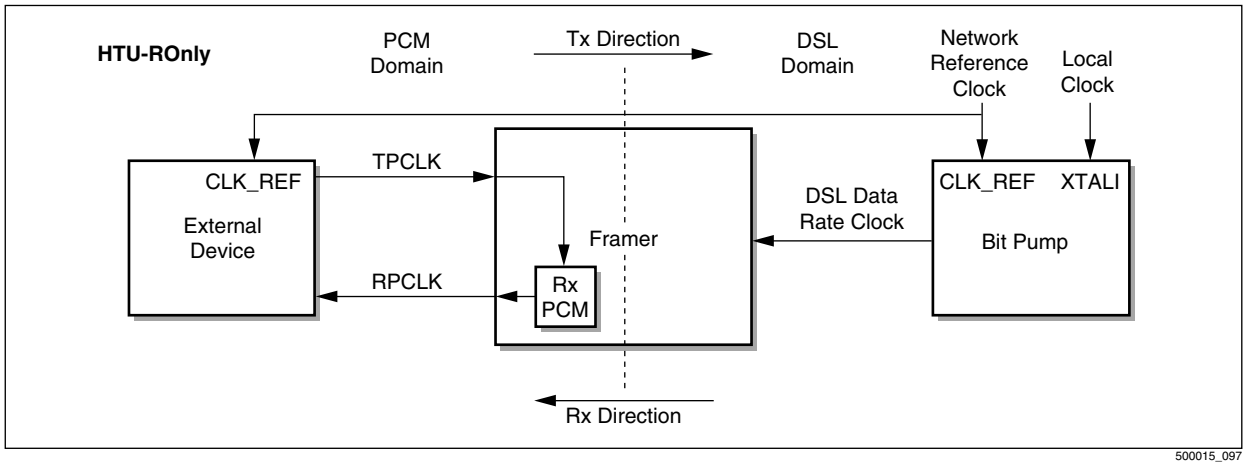


3.2.3.3 Synchronous Network Reference Clock

Both the DSL and PCM clocks are synchronized to a common network reference clock. The TPCLK must be supplied from an external source. The DPLL is disabled.

Figure 3-6 illustrates a simplified block diagram of the synchronous network reference clock mode.

Figure 3-6. Synchronous Mode with Network Reference Clock Block Diagram



3.3 HTU-R Clocking—Network Reference Clock Output

This section describes how the clocking modes are targeted for HTU-R applications. In general, the HTU-R will recover the DSL, PCM, and network references clocks from the incoming DSL line. The PCM clock can operate in loop timed or independent transmit/receive PCM clock modes. The network reference clock output is optional.

NOTE: When operating as an HTU-R, the device can support all HTU-C clocking schemes but the HTU-C modes are not required in most applications.

3.3.1 Independent Transmit/Receive PCM Clocks

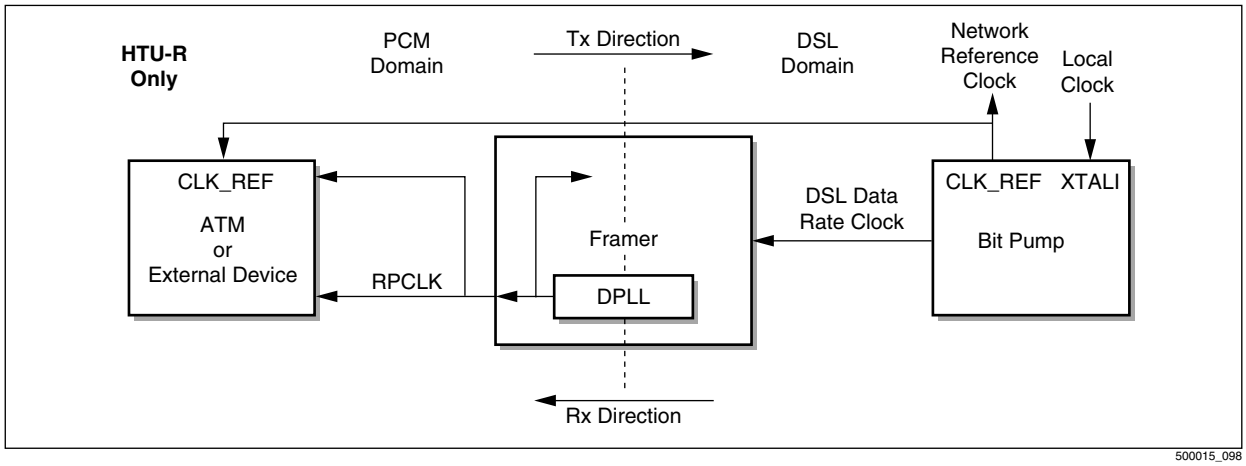
The transmit PCM and receive PCM clocks can operate at independent rates (within the appropriate PPM tolerance). This mode is the same as the HTU-C plesiosynchronous mode, see Section 3.2.1 for details and Figure 3-7.

3.3.2 PCM Loop Timed

This mode takes the DPLL recovered clock (RPCLK) and uses it for both the transmit and receive PCM directions. This mode is applicable in stuffing and non-stuffing modes. This configuration is similar to the synchronous network reference clock master PCM (see Section 3.2.3.2).

Figure 3-7 illustrates a simplified block diagram of the HTU-R PCM loop timed clock mode.

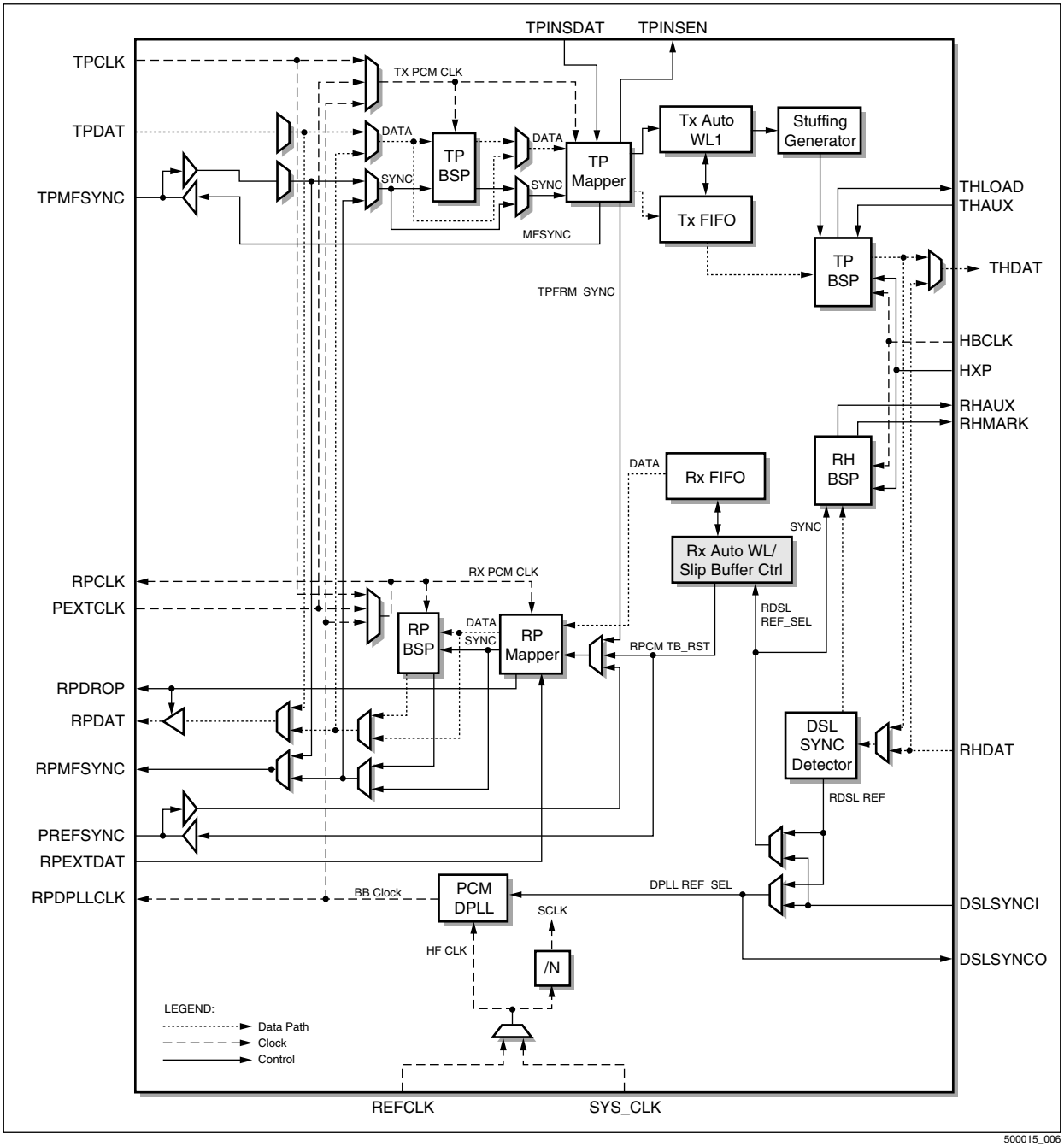
Figure 3-7. Synchronous—PCM Loop Timed Block Diagram



4.0 ZipWireMulti Framer Detailed Description

Figure 4-1 illustrates the detailed block diagram of the DSL Framer block. This section provides a detailed description of the various modules of the DSL Framer block.

Figure 4-1. DSL Framer Detailed Block Diagram



4.1 Distinguishing Features

- Supports all legacy features of Bt8953A and RS8953B
- Compliant with *ETSI RTS/TM-06008* [1]
- 1- 2- or 3- pair T1/E1 ETSI and Bellcore standard application
- PCM interface up to 16 MHz
- ISDN Primary Rate Access (PRA)
- Custom N × 64 over 1 or 2 pairs
- Asymmetric PCM rate and frame format capability
- Various rates of PCM clock recovery (64 kHz to 16 MHz)
- Low jitter (wander) stuffing generator
- Flexible Stuff Bit ID (SBID) mapping, including majority vote decision (HDSL2 applications)
- Three programmable PCM and DSL sync detectors (supports grouped and spread sync word patterns)
- Two programmable PRBS/BER meters to both PCM and DSL sides
- 12 programmable performance monitoring counters (can be used for CRC, BPV, or FEBE error counters)
- Three programmable CRC generators
- Programmable scrambler/descrambler
- Supports variable time slot size (8, 4, 2, or 1) and, therefore, variable PCM custom frequency (N × 64, N × 32, N × 16, and N × 8, respectively)
- Two frames receive PCM slip buffer
- Serial PCM system bus interface up to 16 MHz (AT&T CHI and MITEL ST-SUB)
- UTOPIA Level 2 (8/16 bit mode) interface and ATM Transmission Convergence (TC)
- IDSL frame format capability

4.2 Common Functions

4.2.1 DATA FIFO

The DSL Framer contains two DATA FIFOs, TX_FIFO, and RX_FIFO, used to provide rate buffering between the PCM side data rate and the DSL side data rate. Each FIFO is capable of storing 512 bits (two E1 frames).

4.2.2 Two Frame Receive Slip Buffer

To support channelized voice application, the Framer's transmitter and receiver time base must be aligned, and a receive Slip Buffer compensates any clock differences between the central office and the remote terminal. The receive Slip Buffer depth must be two full frames (E1 or T1) to be able to add or drop a full frame when necessary. The DSL Framer has two receive Slip Buffers, one for each PCM port.

4.3 DSL Section

4.3.1 General DSL Function

The DSL section consists of a transmitter and receiver. The DSL transmitter frames the transmit payload data, inserts the overhead, and scrambles the DSL data. The DSL receiver unscrambles the DSL data and removes the overhead.

4.3.1.1 CRC Generator

A generic CRC generator with selectable taps (up to 7th order) is implemented. CRC calculation can be corrupted for debugging purposes. This mode simply inverts the CRC calculation.

4.3.1.2 Scrambler/ Descrambler

The Scrambler/Descrambler operation can be bypassed for debugging.

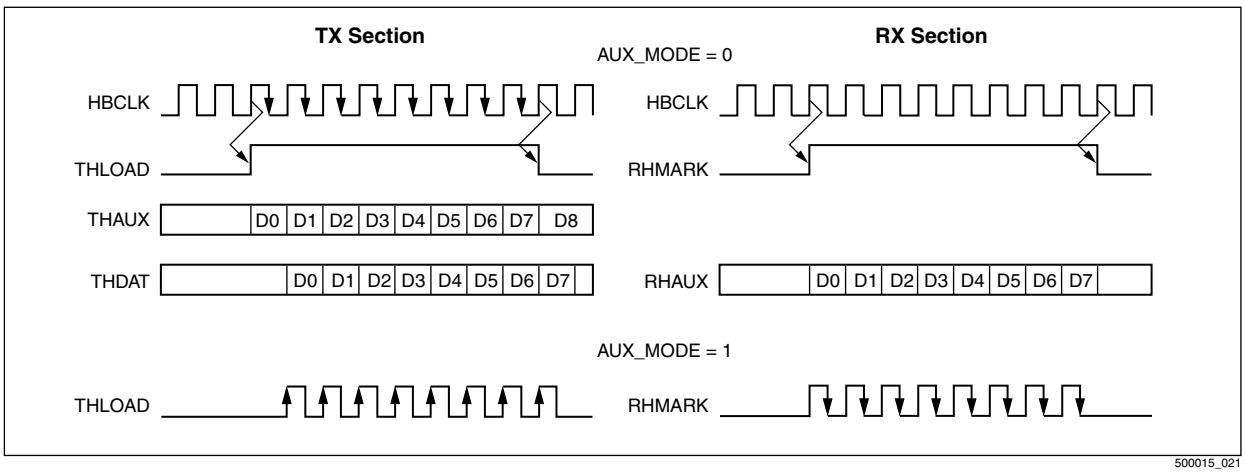
4.3.1.3 Auxiliary Channel

The DSL Auxiliary Channel (THAUX, RHAUX) provides an alternate source of DSL payload. This channel supports any payload size and optionally can function as an alternate source for the Z-bits or any other selected overhead.

The Auxiliary Channel interface has two operational modes. In the first mode, THLOAD and RHMARK signals simply mark high during auxiliary input mode. The second mode generates gated clock-in pins THLOAD and RHMARK during Auxiliary mode to clock the serial device directly. This mode prevents additional glue logic in the interface between DSL Framer and the serial device.

Figure 4-2 illustrates the DSL Auxiliary Channel timing.

Figure 4-2. DSL Auxiliary Channel Timing



4.3.1.4 RX DSL Reference Phase Measurement

While working in Multi-Pair configuration, the DSL Framer can measure the receiving DSL phase difference between two pairs. This phase is used mainly to determine the delta delay between two DSL channels in point-to-multi-point application (and can be also used for debugging or link delay measurement).

4.3.2 DSL Receiver Functionality

4.3.2.1 DSL Sync Detector (DSD)

The DSL Sync Detector (DSD) acquires and maintains synchronization of the DSL.

To support the wide variety of frame formats, the DSD is designed in a flexible way which provides the following capabilities:

- Synchronized to any grouped bit sync pattern up to 16 bits long.
- DSL frame size (nominal) can be up to 2^{16} (65536) bits long.
- Stuff size can be 2, 4, 6, or 8 bits. For applications that do not need stuff bits (HDLC applications), the DSD can search for the sync word without searching in variable frame length, but instead, search for a fixed location.

NOTE: The default configuration is adequate for most standard applications. For some custom applications, this option requires modifying the low-level DSL Framer code.

4.3.2.2 Tip/Ring Reversal Detection

During 2B1Q mode, Tip/Ring reversal is automatically detected and corrected by the DSD.

In HDSL2 applications, the Tip/Ring reversal cannot be detected by the DSL Framer due to the non-symbol alignment nature of the DSP operation. In this case, the DSP is responsible for detecting and correcting Tip Ring Reversal.

4.3.3 DSL Transmitter Functionality

4.3.3.1 Stuffing Generator

The stuffing generator synchronizes the DSL frame period to the PCM frame period by adding 0, 2, 4, 6, or 8 STUFF bits (0,4 in HDSL1 application) to the DSL frame period.

The stuffing generator can be bypassed for nonvariable frame length application and as an additional debugging tool.

4.4 PCM Section

The PCM section (receiver and transmitter) is composed of two major blocks, the PCM mapper and the Layer 3 Framer.

The PCM mapper functions as a formatter which maps or extracts PCM payload data into or out of the DSL channel through the FIFO. Additionally, the mapper can override the data with data from a data bank register or generate a PRBS sequence.s

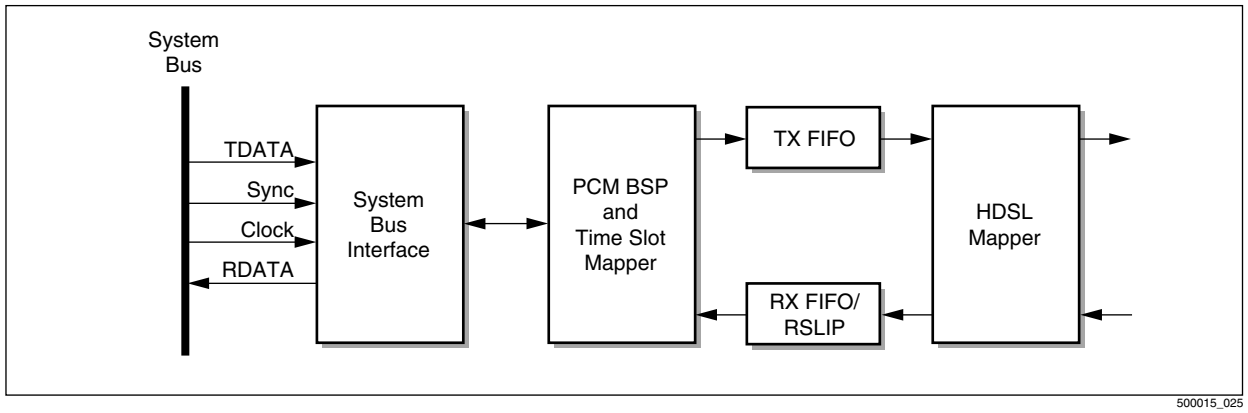
The Layer 3 Framer synchronizes to a PCM frame or multiframe, extracts or inserts overhead data out of or into the PCM Frame, and checks for any block errors (such as CRC).

4.4.1 PCM Interface

In PCM highway interface applications of 8E1/8T1, the DSL framer interface is compliant with AT&T CHI and MITEL ST-BUS interfaces up to 16 MHz. This interface enables multi-pair configuration or any kind of PCM aggregation mode, which allows data from different sites to be aggregated onto the same PCM bus.

Figure 4-3 illustrates the integrated slip buffer and system bus interface.

Figure 4-3. Integrated Slip Buffer and System Bus Interface



4.4.2 General PCM Functions

4.4.2.1 CRC Generator

The PCM section contains two generic CRC generators, functionally identical to the one located on the DSL side (see Section 4.3.1.1 for more details).

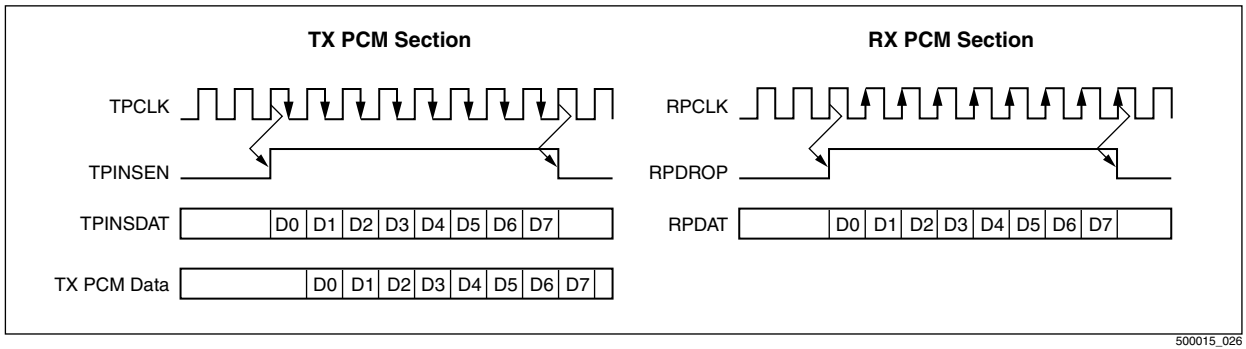
The PCM CRC calculation can be corrupted for debugging purposes. This mode simply inverts the CRC calculation. In addition, the CRC generator can be bypassed or recalculated.

Any CRC computation format can be generated. CRC computation can be disabled or enabled. There is the capability to replace any bit in the frame with either a 0 or 1 for CRC computation purposes. This capability allows for support of any CRC operation method.

4.4.2.2 Insert/Drop

An alternate PCM source feeds into the PCM formatter, using TPINSEN, TPINSDAT, and RPDROP pins (see Figure 4-4).

Figure 4-4. Insert/Drop Timing Diagram



4.4.2.3 Overhead Handling

The PCM transmitter and PCM receiver can handle up to 24 OverHead (OH) bytes that can function as Sa-bits, E-bits, and A-bits for E1 applications. They can be used to generate any in-band management.

NOTE: This option requires modifying the low-level DSL Framer code.

4.4.2.4 E1 Grooming

To support the E1 P2MP application, it is necessary to groom Channel Associated Signaling (CAS) from different sites. Each remote site has a different PCM Frame sync that needs aligning in the central site.

4.4.2.5 MF Phase Measurement

During E1 point-to-multi-point application, PCM multi-frame phase measurement between TPMFSYNC and RPMFSYNC pins, with respect to internal transmit MF Sync (MFSYNC), is necessary at the remote site to compensate for misalignment between different remote sites. This phase can then be used to internally align the DSL transmits frame to the PCM frame boundary in each site. It can provide the value of each site to the central office and align the receive channel signaling to the receive E1 MF.

4.4.3 PCM Receiver

The major tasks of the PCM Receiver are to the following:

1. Generate RX PCM time base aligned with the DSL reference (WL delay apart).
2. Assemble ongoing PCM frame using flexible RX PCM Mapper Table. Major tasks of this table are to:
   a. Assemble RX PCM frame from selectable sources: DSL payload, PRBS sequence, DATA BANK 1, 2 or 3, Signaling table (Grooming Mode), and RPEXTDAT input pin.
   b. Enable BER meter per time slots basis.
   c. Asserts RPDROP pin to signify specific TSs in RPDAT output.
   d. Insert external PCM data (in RPEXTDAT input) to the receive PCM Frame. Used in multi-pair configuration.
3. Generate receive user interface SYNC signals such as RPMFSYNC and PREFSYNC (multi-pair configuration PCM time-base sync).
4. Synchronize to any Layer 3 Frame/MF.
5. Check CRC (selectable) and computes CRC on the final ongoing frame.
6. Extract overhead bits.
7. Provide capability to override each overhead bit by the MPU.

4.4.4 PCM Transmitter

The major tasks of the PCM transmitter are to

1. Generate Tx PCM time base aligned with the incoming Frame/Multiframe sync.
2. Map PCM Frame to the TX_FIFO using TX PCM Mapper Table.
3. Enable BER meter per time slots basis.
4. Inserts alternate PCM channel, using TPINSDAT, TPINSEN pins.
5. Synchronizes to any Layer 3 Frame/MF.
6. Check CRC (selectable) and compute CRC on the final ongoing frame.
7. Extract overhead bits.
8. Provide capability to override each overhead bit by the MPU.

4.4.4.1 PCM Sync Detector

The Sync Detector can synchronize to any sync pattern, grouped or spread, up to 16 bits long. This capability allows the DSL Framer to synchronize to E1, T1, or any other frame. This requires modifying the low-level DSL Framer code.

4.4.5 Primary Rate Access (PRA)

This Section describes the Primary Rate Access (PRA) software feature.

4.4.5.1 PRA Distinguishing Features

The following provides a list of the PRA software features. The list is broken into the ZipWireMuli and host processor responsibilities. The software can terminate and insert the PRA functionality in both the transmit and receive direction.

ZipWireMuli Responsibilities

- Align E1 frame in DSL Frame
- Automatic Multi-frame Alignment (or use external sync reference)
- Output 2mS Multi-frame in receive direction
- Automatic CRC detection and insertion
- Automatic CRC-4 error indication bits (E bits)
- Transmit remote alarm indication bits (A bits) pattern
- Process spare bits (Sa4, Sa5, Sa6, Sa7, Sa8) up to message level

Host Processor Responsibilities

- Process spare bits (Sa4, Sa5, Sa6, Sa7, Sa8) messages
- Set remote alarm indication bits (A bits) via API command

4.4.5.2 PRA Overview Description

Figure 4-5 illustrates on overview of the PRA data path and terminology. The PRA functionality monitors and manipulates the overhead bits found in time-slot 0 on an E1 frame; which includes CRC-4, E-bits, A-bits, and Spare (Sa) bits. In addition, the PRA block can either automatically lock onto the incoming multiframe boundary or except the multiframe from an external source.

The PRA functionality can be performed in both the transmit and receive directions. The transmit direction is defined as the data path from the PCM interface towards the DSL interface. The receive direction is defined as the data path from the DSL interface towards the PCM interface. Each direction consists of a monitor and generator block. The PRA functionality is identical in each direction but each direction has independent control.

In general, there are three possible ways to control the various overhead bits – transparent, automatic, and manual. Refer to each overhead bit definition to determine which modes apply.

- Transparent—the generator block outputs the overhead bits unaltered from the monitor block.
- Automatic—the generator block will automatically determine and output the appropriate overhead bits value. The overhead bits are updated every multiframe sync.
- Manual—the generator block outputs the overhead bits based on an API command received from the host processor. The generator block will continuously output the new value starting on the subsequent multiframe sync.

Figure 4-5. PRA Overview

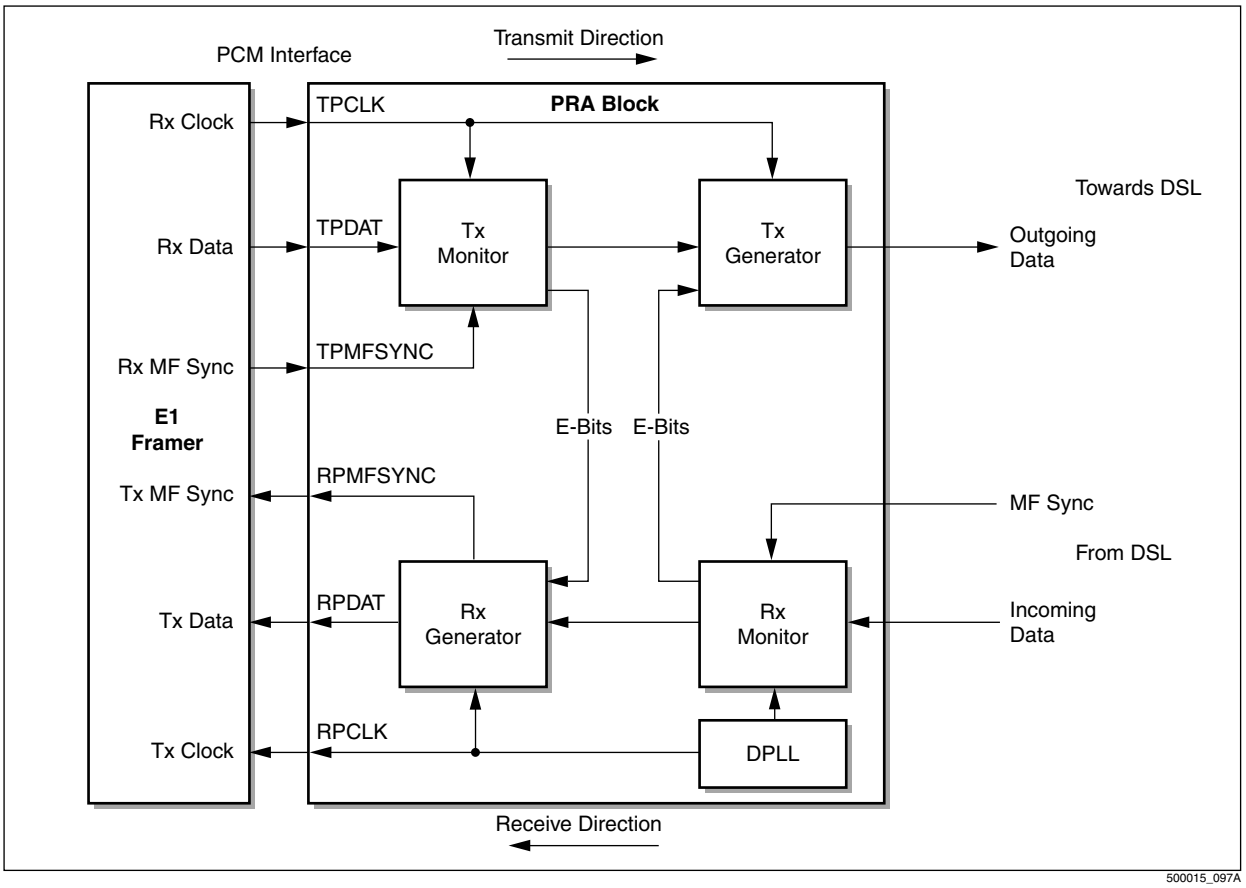


Table 4-1 lists the time-slot 0 bit definition for the multi-frame. The multi-frame is broken into 2 sub-multiframes. The CRC-4 is calculated on a sub-multiframe boundary and inserted into the next sub-multiframe boundary.

The grayed bits define the multi-frame alignment pattern, see Section 4.4.5.3 for details.

Table 4-1. Multi-frame Structure

Sub Multi-Frame	Frame #	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	1	C1	0	0	1	1	0	1	1
	2	0	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	3	C2	0	0	1	1	0	1	1
	4	0	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	5	C3	0	0	1	1	0	1	1
	6	1	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	7	C4	0	0	1	1	0	1	1
	8	0	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
2	9	C1	0	0	1	1	0	1	1
	10	1	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	11	C2	0	0	1	1	0	1	1
	12	1	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	13	C3	0	0	1	1	0	1	1
	14	E1	1	A	Sa4	Sa5	Sa6	Sa7	Sa8
	15	C4	0	0	1	1	0	1	1
	16	E2	1	A	Sa4	Sa5	Sa6	Sa7	Sa8

4.4.5.3 Multi-Frame Alignment

In the transmit direction, the ZipWireMulti can either self align to the multiframe or accept an external 2mS multiframe reference input. In the receive direction, the ZipWireMulti can either self align to the multiframe or use the DSL 6mS reference.

Self Alignment

In Table 4-1, the 13 grayed bits define the multi-frame alignment pattern.

4.4.5.4 CRC-4 Processing

See Section 4.4.5.5 for the CRC-4 formula.

CRC-4 Generator

The CRC-4 generator can be configured to transparent or automatic mode. In transparent mode, the CRC-4 is outputted unaltered from the monitor block. In automatic mode, the CRC-4 is re-calculated based on the E1 data stream.

CRC-4 Monitoring

The CRC-4 monitoring block calculates the incoming E1 data stream and compares it to the CRC-4 result. The result of this check is referred to as the E-bit, Section 4.4.5.5 for details.

In addition, the monitor block maintains a 16-bit CRC-4 error counter that wraps around on full count. The CRC-4 error counter is always available when PRA is enabled.

4.4.5.5 CRC-4 Error Indication Bit (E-bit) Processing

The CRC-4 Error indication bits are used to indicate to the far-end unit that a CRC-4 error was detected in the incoming data stream.

The E1 and E2 error indicator bits indicate the status of the previously received multiframe CRC-4 status. The E1 bit indicates the previous sub-multiframe 1 status while the E2 bit indicates the previous sub-multiframe 2 status.

The E bit will be set to 1 when no CRC-4 error is detected and will be set to 0 when a CRC-4 error is detected. Both E bits will be set to 0 until both basic and multiframe alignment are established.

CRC-4 Error Bit (E-bit) Generator

The E-bit generator can be configured to transparent, manual, or automatic mode.

In transparent mode, the E-bit is outputted unaltered from the monitor block in the same direction. For example, the transmit direction generator E-bit will match the transmit direction monitor's E-bit.

In manual mode, the E-bit is determined from the host processor via an API command.

In automatic mode, the E-bit is determined based on the opposite direction monitor's CRC-4 result. For example, the transmit direction generator E-bit will be set to 0 if the receive direction's monitor detected a CRC-4 error.

Monitor's CRC-4 Check	Generator's E-Bit Value
Error	0
No error	1

CRC-4 Error Bit (E-bit) Monitor

The E-bit monitor block checks the value of the E-bits. If the corresponding direction is in transparent mode, the incoming E-bit value is output on the next multiframe. If the corresponding direction is in manual or automatic mode, the incoming E-bit value is ignored.

In all cases, the monitor block maintains a 16-bit E-bit error counter that wraps around on full count. The E-bit error counter is always available when PRA is enabled.

Remote Alarm Bit (A-bit) Processing

The Remote Alarm Bit (A) indication bits are used to indicate an alarm condition to the far-end unit. According to the standards, the A-bit will be set to 0 in an undisturbed operation condition and will be set to 1 in an alarm condition.

A single control bit controls the A-bit generator and monitor for each direction. The A-bit generator and monitor can be configured to transparent or manual mode.

In transparent mode, the generator outputs the A-bit value unaltered from the monitor block.

In manual mode, the generator A-bit value is determined from the host processor via an API command and is continuously output starting on the next multiframe. The monitor block will notify the host processor whenever there is a change in the received A-bit pattern and all 8 A-bit are identical.

4.4.5.6 Spare Bit (Sa4 to Sa8) Processing

The Spare Bits (Sa) are provided for user-defined messages. The 5 Sa bytes have identical functionality but have independent control.

For each Sa byte, a single control bit controls the Sa generator and monitor for each direction. The Sa generator and monitor can be configured to transparent or manual mode.

In transparent mode, the generator outputs the Sa value unaltered from the monitor block.

In manual mode, the generator Sa value is determined from the host processor via an API command and is continuously output starting on the next multiframe. The monitor block will notify the host processor whenever there is a change in the received Sa pattern and the pattern is received in 8 consecutive multiframes.

4.5 ATM-TC and UTOPIA Level 2 Interface

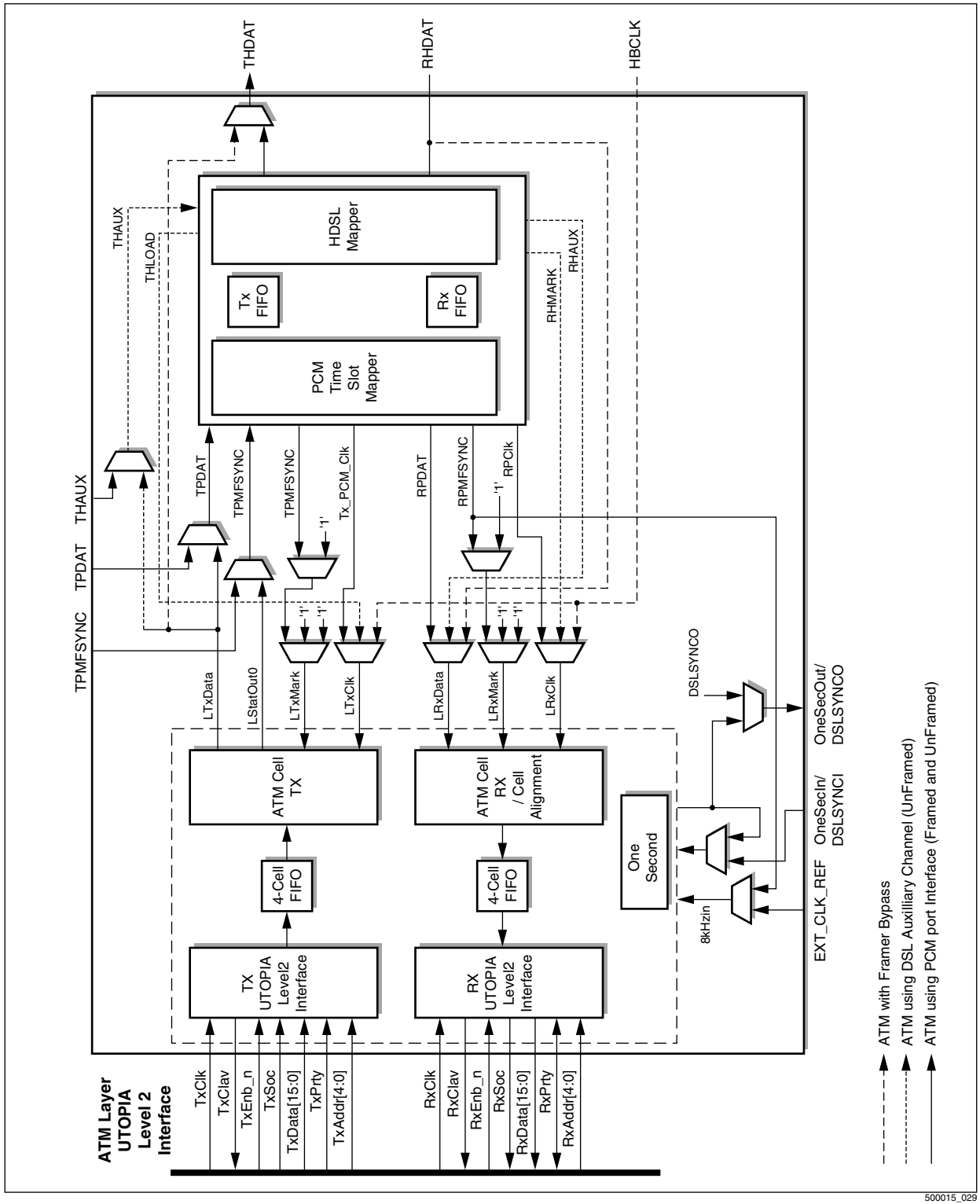
An ATM-TC (Transmission Convergence) and UTOPIA Level 2 (or Level 1 as an option) interface are fully supported by the DSL Framer, compliant with ATM Forum standards. This block is a slice of one port out of the octal ATM-TC PHY Device (RS8228).

The ATM-TC and UTOPIA interface support the following features:

- UTOPIA level 2, Multi-PHY addressing cell bus interface supports up to 31 PHYs.
- 50 MHz UTOPIA interface maximum clock rate.
- Supports all ATM physical layer processing functions found in ATM Forum Cell-Based Transmission Convergence Sublayer specification.

Figure 4-6 illustrates the Integrated One Port of RS8228 with xDSL Framer.

Figure 4-6. Integrated One Port of RS8228 with xDSL Framer



4.6 Test and Diagnostics

4.6.1 Performance Monitoring

The DSL Framer supports up to 12 performance monitoring counters, divided equally into three sections.

Each performance-monitoring counter can function as a CRC error counter for the Sever Error Second (SES) indicator, Far End Block Error (FEBE) counter, Bipolar Violation (BPV) error counter, or any other necessary performance indicator counter. The Receive DSL, Receive PCM, and Transmit PCM support up to four performance monitoring counters.

4.6.2 PRBS and BER Meter

The DSL Framer has four PRBS/BER meter modules, supporting BER measurement towards the DSL, PCM, and NB sides. It can operate independently. The following description focuses on the PCM port; however, the same description applies to the NB port.

TP_PRBS (TNB_PRBS) and RP_BER (RNB_BER) function as BER meters towards the HDSL side. The RP_PRBS and TP_BER function as BER meters towards the PCM side.

The PRBS sequence can override TPDAT and RPDAT per time slot basis, and achieve any framed or unframed test pattern examination. The PRBS pattern is programmable and selected for both RP_BER and TP_BER by PRBS_TAP_[2:0] registers, indicating up to 23rd-order PRBS (Tap[23:0]):

Example:

For a PRBS pattern of $2^{15} – 1$, the polynomial is $x^{15} + x^{14} + 1$.

For a PRBS pattern of $2^{23} – 1$, the polynomial is $x^{23} + x^{18} + 1$.

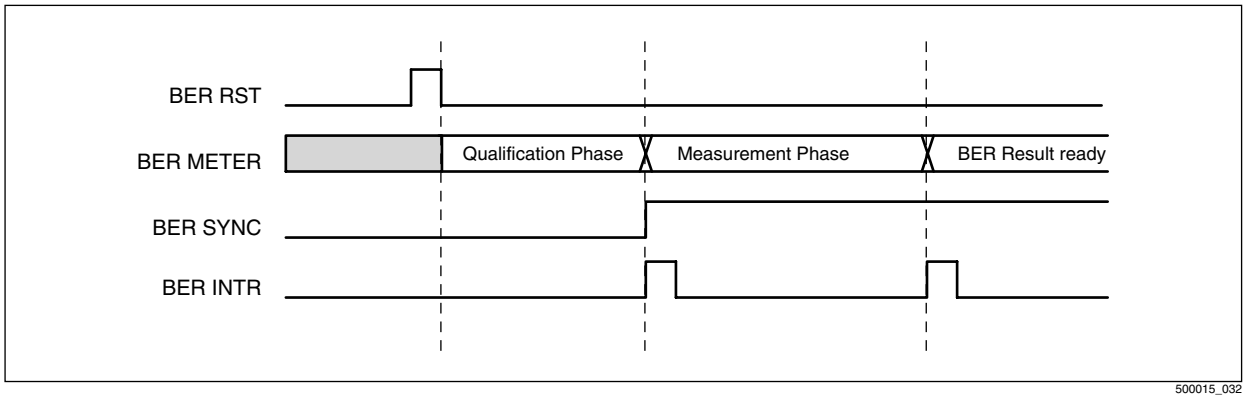
For a QRSS $2^{20} – 1$ pattern (polynomial – $x^{20} + x^{17} + 1$), 14-bit 0 suppression is implemented.

The TP_BER and RP_BER sequence can be inverted.

The constant value per time slot basis can override TSER and RSER instead of PRBS. The MPU configures *BER_SCALE* to specify the test measurement interval from a range of 2^{21}–2^{31} bit length.

Figure 4-7 illustrates the BER Measurement timing.

Figure 4-7. BER Meter Timing

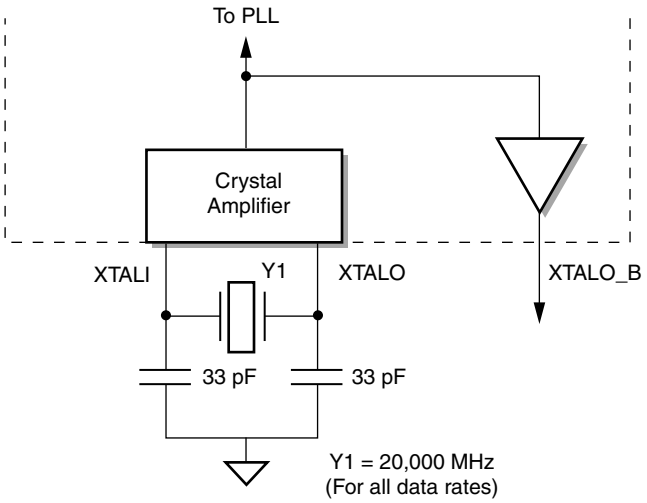


5.0 Hardware Interfaces

5.1 ZipWireMulti Clocks

The ZipWireMulti DSP block generates the DSP, Microprocessor, AFE, and Framer reference clocks. The ZipWireMulti Framer has a DPLL to generate the PCM receive clock.

Figure 5-1. Crystal Interface



NOTE(S):
The value of the capacitors may vary, depending on the PCB capacitance.

500015_098A

Table 5-1. Crystal Specifications

Parameter	Value
Nominal Frequency	16.384 MHz
Frequency Tolerance at 25 °C	± 10ppm
Temperature Frequency Stability	± 10 ppm
Aging	± 10 ppm over 10 years
Load Capacitance	15.5 pF
Max Shunt Capacitance	7 pF
Max Drive Level	200 µW
Max ESR	25 Ω
NOTE(S): Individual Frequency tolerance, temperature frequency stability, and aging requirements can vary as long as the total tolerance is less than +/– 32 ppm.	

Figure 5-2. Direct Clock Connection

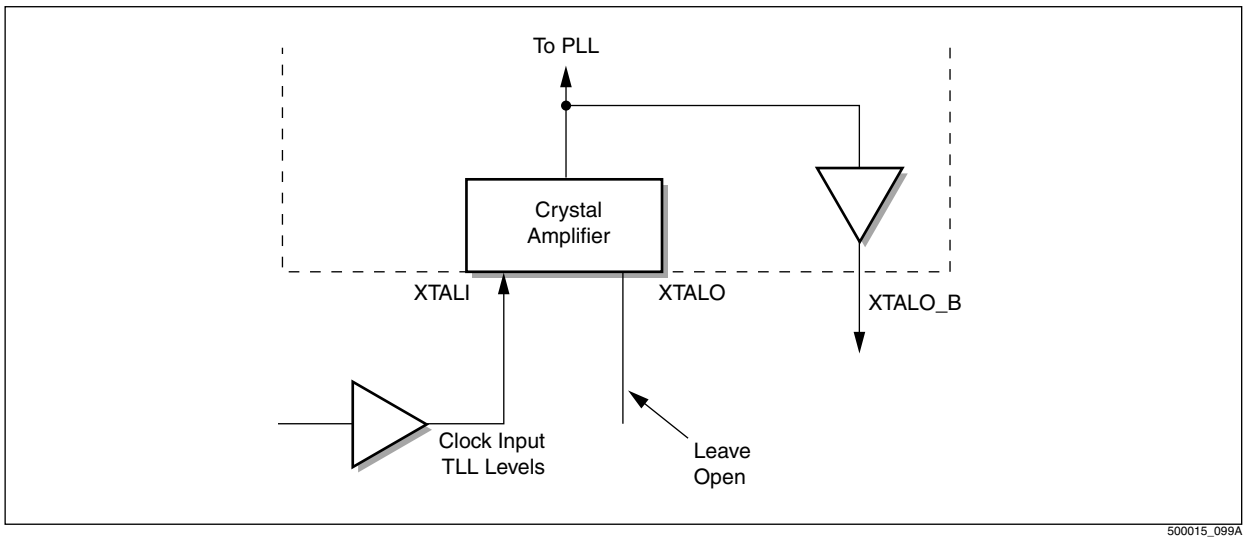


Table 5-2. ZipWireMulti Clocks

Clock or Node	Frequency	Description
XTALI	16.384 MHz	External crystal or clock input
XTALO	16.384 MHz	Crystal output
XTALO_B	16.384 MHz	Buffered crystal output
AFE_CLK	21.5–26.5MHz	AFE clock reference

5.2 Host Bus Interface

This section specifies the functional characteristics of the host bus interface for the CX28985 which implements the interface between the core processor and an external host processor. On the host bus side, the purpose of the host interface is to provide a simple external interface that can be connected to as many of the popular microprocessor buses as possible. The host interface operates on two independent clock domains, the host clock, and the ARM clock. Control signals that cross clock domains are synchronized through two levels of synchronization flops.

5.2.1 Interface Description

The host interface hardware consists of two FIFO buffers (FIFOs) and eight mailbox registers. It provides one FIFO for the outbound direction and one FIFO for the inbound direction. Both inbound and outbound FIFOs have separate FIFO size registers, which indicate the current FIFO depth, and FIFO threshold registers, which can be programmed by the host.

The outbound FIFO provides status bits, with interrupts if required, when the outbound FIFO is either empty or below the programmed threshold. The inbound FIFO provides status bits, with interrupts if required, when the inbound FIFO is either full, or above the programmed threshold.

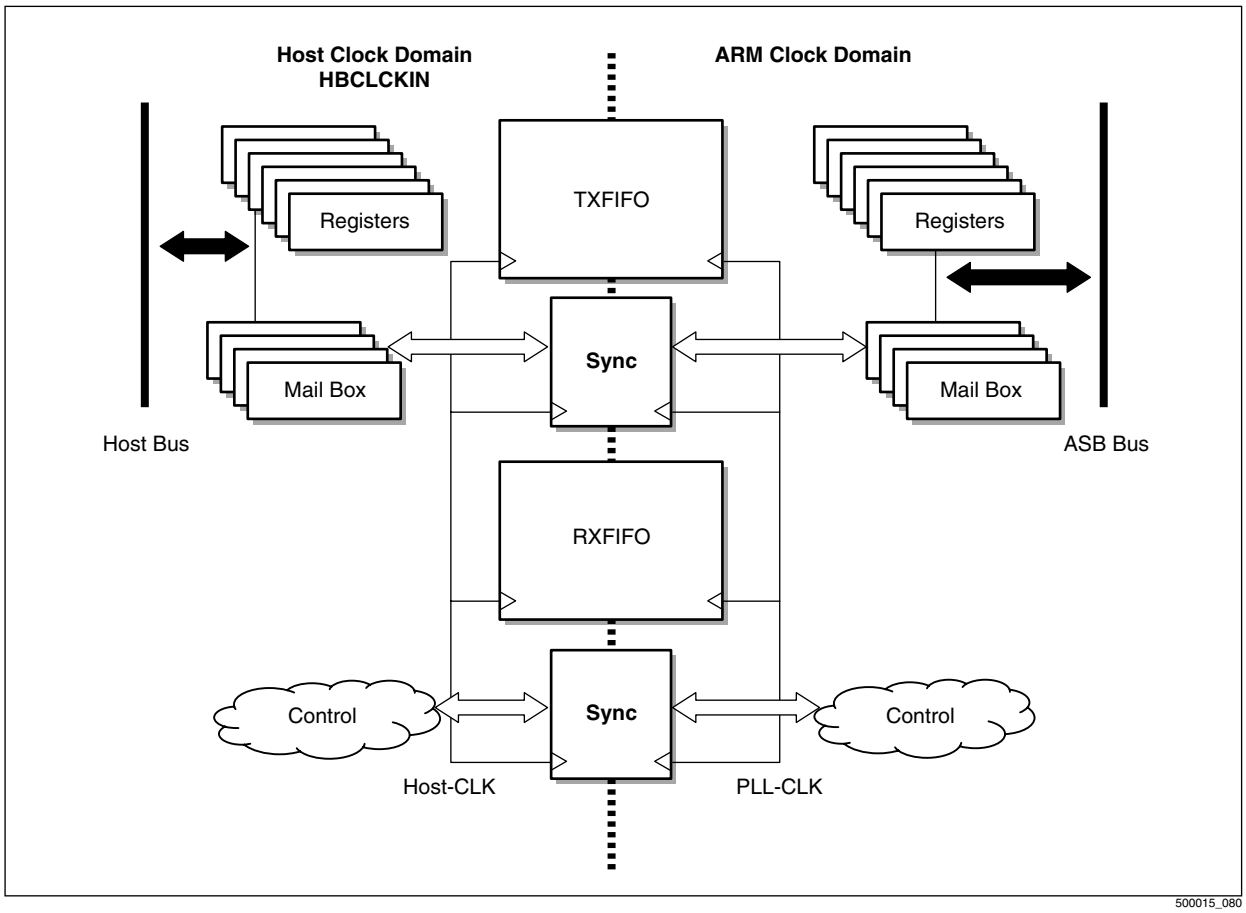
The host interface provides eight mailbox registers, four designated for the inbound direction, and four designated for the outbound direction. The CX28985 provides a hardware handshake mechanism to simplify the passing of messages through the mailbox registers.

For outbound messages, a status bit (with interrupt if required) indicates that the host can send a new message. The host must clear this bit (by writing to a control register) when a new message is to be sent, and then write the message to the outbound mailbox registers. Writing to outbound mailbox register 3 triggers the hardware handshake, which tells the CX28985 that a new message has been written. When the CX28985 has read the message, it sets its acknowledge control bit, which triggers the hardware handshake (by setting the status bit) to inform the host once again that it can send a new message.

For inbound messages, the same handshake is used, only the host now sees the transfer from the opposite side. The host sees a status bit (with interrupt if required) when a new message is in the inbound mailbox registers. The host reads the message, then sets its acknowledge control bit, which triggers the hardware handshake to inform the CX28985 that another inbound message can be sent.

Figure 5-3 illustrates the Host Interface Block Diagram.

Figure 5-3. Host Interface Block Diagram



5.2.2 Host Bus Signals

Table 5-3. Host Bus Interface Signals

Signal Name	Type	Description
HBA[2:8]	I	Host Bus Address. HBA[1] is not used because all host bus registers are 16-bit registers aligned at the 32-bit boundary, and FIFO has 16-bit access capability. HBA[8] is used to decode register or FIFO access (H to access FIFO and L to access registers).
HBADDWT#	I	Host Bus Add Wait State. HBADDWT# = L adds one HBCLKIN to the host bus access in Burst mode, HBBURSTEN# = L, so each data is transferred in 2 HBCLKIN instead of every HBCLKIN clocks. In Non-burst mode, this signal does not have effect on the data transfer.
HBBE1#/HBA[0], HBBE0#/GND.	I	Host Bus Byte Enable. In 16-bit bus mode, HBBE1# = L indicates D[15:8] byte is valid on the current bus cycle, and HBBE0# = L indicates D[7:0] byte is valid on the current bus cycle. In 8-bit bus mode, HBBE1# should be connected to host bus HBA[0], and HBBE0# should be grounded.
HBBURSTEN#	I	Host Bus Burst Enable. HBBURSTEN# = L to enable burst access to the CX28985, so multiple data can be transferred on single pulse of HBCS# = L. When HBBURSTEN# = H, the burst is not enabled and each HBCS# = L pulse has only one data transfer, either 8- or 16-bit.
HBCLKIN	I	Host Bus Clock IN. The reference clock of the host bus; when HBINVCLK# = H, the host bus signals are clocked at the rising edge of the HBCLKIN; when HBINVCLK# = L the host bus signals are clocked at the falling edge of the HBCLKIN.
HBCS#	I	Host Bus Chip Select. HBCS# = L selects the CX28985 to participate on the current bus cycle, and HBCS# = H tells the chip not to participate on the current bus cycle.
HBD[0:15]	B	Host Bus Data. HBD[15] is the most significant bit and HBD[0] the least significant bit. In 8-bit bus mode, only HBD[7:0] signals are used, and HBD[15:8] signals should be pulled up individually (many CX28985 can share the same pull-up resistors).
HBINT#	O	Host Bus Interrupt.The attention signal to the host processor. It is active low and open drain so many chips can tie it together. An external pull up resistor is needed.
HBINVCLK#	I	Host Bus Invert Clock. When HBINVCLK# = H the host bus signals are clocked at the rising edge of the HBCLKIN, when HBINVCLK# = L the host bus signals are clocked at the falling edge of the HBCLKIN. (Note: this is the strapping signal and should not be changed after system reset.)
HBWRITE#	I	Host Bus Write. When HBWRITE# = H, the host bus cycle is a read cycle and data is transferred from CX28985 to the host processor, and when HBWRITE# = L, the bus cycle is a write cycle and the data direction is opposite from the read cycle.

5.2.3 Host Processor Bus Widths

The CX28985 host bus interface is designed for different sized bus host processors. Both 16- and 32-bit host processors should use the 16-bit interface mode (default), while an 8-bit host processor should use the 8-bit interface mode. After the reset, the 8-bit host processor should write 0x01 to the HBCFG register to configure CX28985 into 8-bit mode operation. HBCFG register is an 8-bit register accessible either 8-bit or 16-bit interface mode through data bus HBD[7:0].

Table 5-4 lists the recommended hardware connection with different host processors.

Table 5-4. Recommended Hardware Connections

Cx28985 Signals	8-Bit Bus Host Processor	16-Bit Bus Host Processor	32-Bit Bus Host Processor
HBA[8:2]	address[8:2]	address[8:2]	address[8:2]
HBBE1#/HBA[0]	address[0]	byte enable for data[15:8], active low	byte enable for data[15:8], active low
HBBE0#/GND	GND	byte enable for data[7:0], active low	byte enable for data[7:0], active low
HBD[15:8]	individually pull up	data[15:8]	data[15:8]
HBD[7:0]	data[7:0]	data[7:0]	data[7:0]

5.2.4 Host Bus Access Description (8-bit or 16-bit Interface)

The CX28985 host bus can be configured to run in Burst or Non-burst mode. When HBBURSTEN# = L, Burst mode is enabled, and multiple data can be transferred on a single HBCS# = L pulse (Figures 5-4 and 5-5). When HBBURSTEN# = H, Non-burst mode is enabled, and only single data is transferred per each HBCS# pulse (Figures 5-6 and 5-7).

For system designs with HBCLKIN frequencies in > 25 Mhz, it may be required to add wait state on the burst transfers. HBADDWT# can be used to add one more clock on each data transfer in Burst mode. In Burst mode, when HBADDWT# = H, data is transferred on each HBCLKIN clock when HBCS# is asserted. When HBADDWT# = L, the data is transferred on every two HBCLKIN clocks when HBCS# is asserted (Figures 5-8 and 5-9).

HBADDWT# does not have to be statics during a burst access. For example, if the system designer found the first data of the burst read access is the constraint of the system design, he can make HBADDWT# signal invert of HBCS# signal and thus make only the first burst data 2 clock access while the second data on 1 clock access (Figure 5-11).

In Non-burst mode, a single data is transferred at the end of HBCS# pulse. Whether the HBCS# pulse is one or hundreds of HBCLKIN cycles long, the data is written to the chip only at the last HBCLKIN edge of the HBCS# pulse. On the read cycle, the data is driven as long as HBCS# is asserted.

In Non_burst mode, a HBCS# cycle (high and low) of the back-to-back accesses must be at least 3 HBCLKIN cycles long. In other words, HBCS# cycle should consist of at least 2 high cycles followed by 1 low or 1-cycle-high followed by 2-cycles-low in the back-to-back accesses. Remember that a Non_burst cycle only ends on the second clock (clock cycle 6 of the Figures 5-6 and 5-7) after HBCS# is disasserted, and the next access starts only after the previous access ends, whether HBCS# is asserted before or not.

Some use falling edge of the clock as the reference. HBINVCLK# signal can be used to facilitate the design with these microprocessors. When HBINVCLK# = H, the host bus interface circuit uses the rising edge of HBCLKIN as the reference, and the write data is written on the rising edge of HBCLKIN. When HBINVCLK# = L, the falling edge of HBCLKIN is used as the reference, and the write data is written on the falling edge of HBCLKIN (Figure 5-10).

Figure 5-4. Host Bus Burst Read

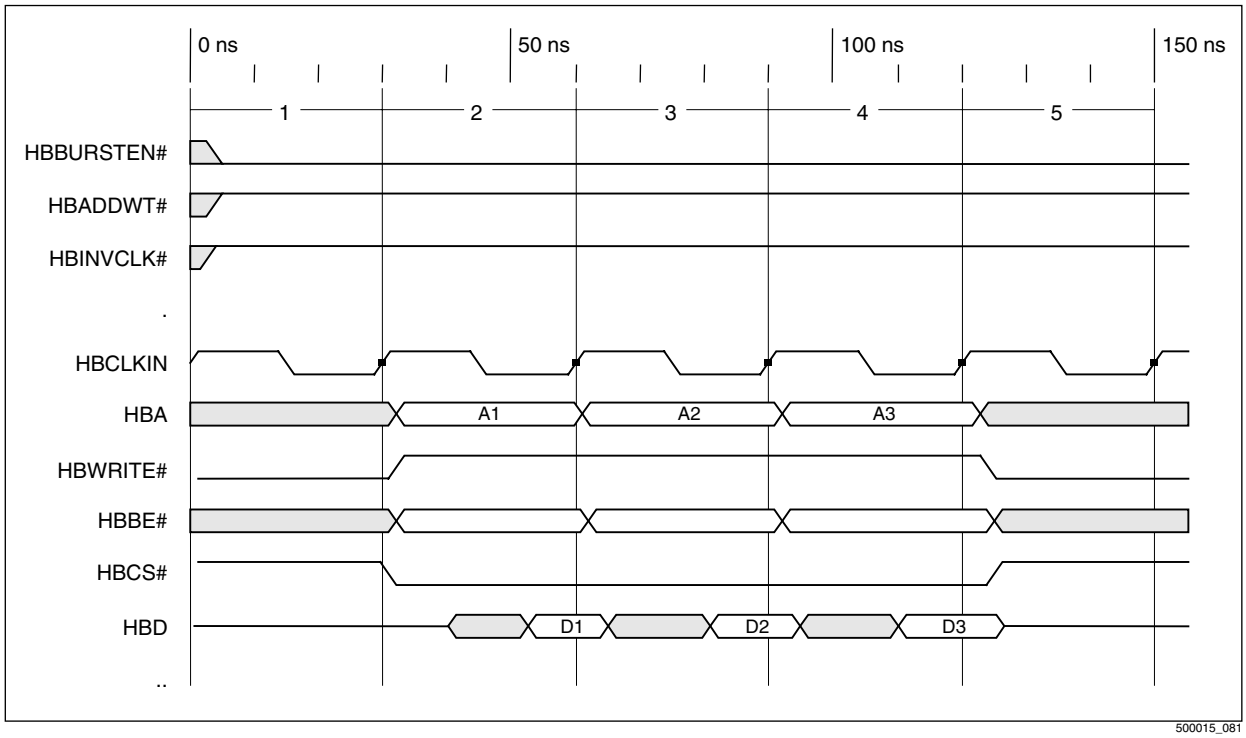


Figure 5-5. Host Bus Burst Write

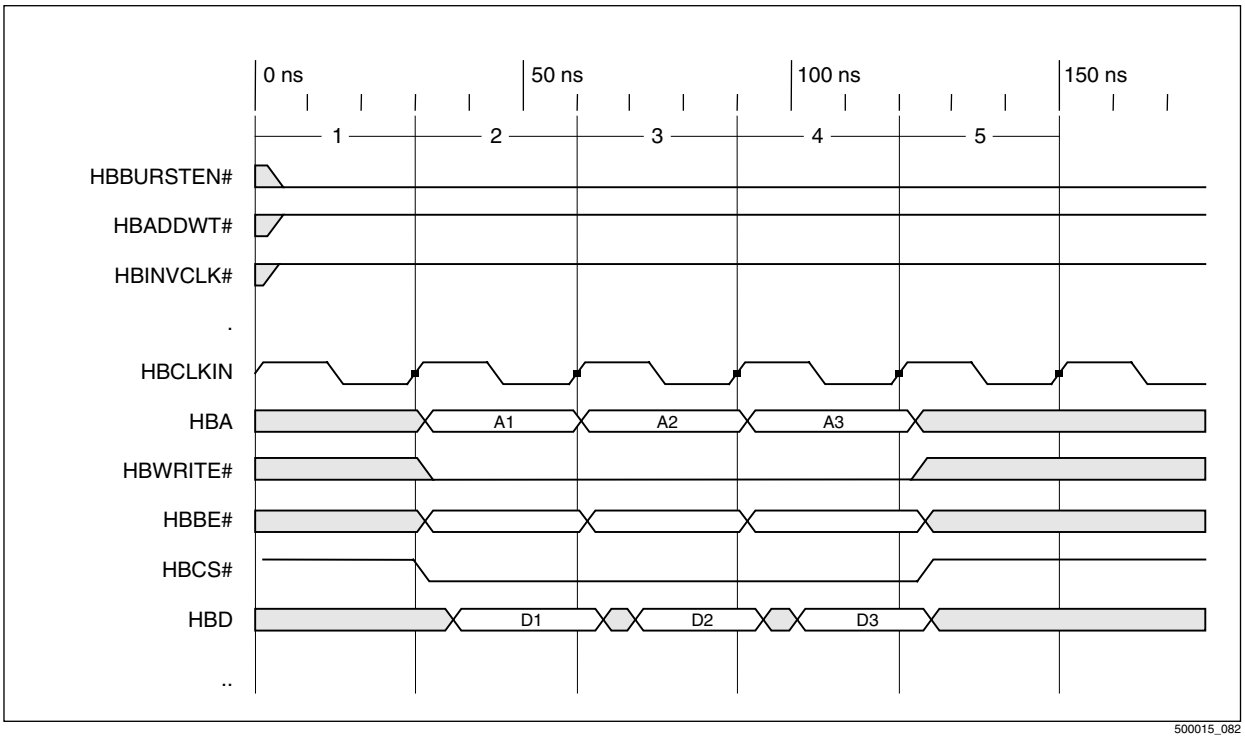


Figure 5-6. Host Bust Non-burst Write

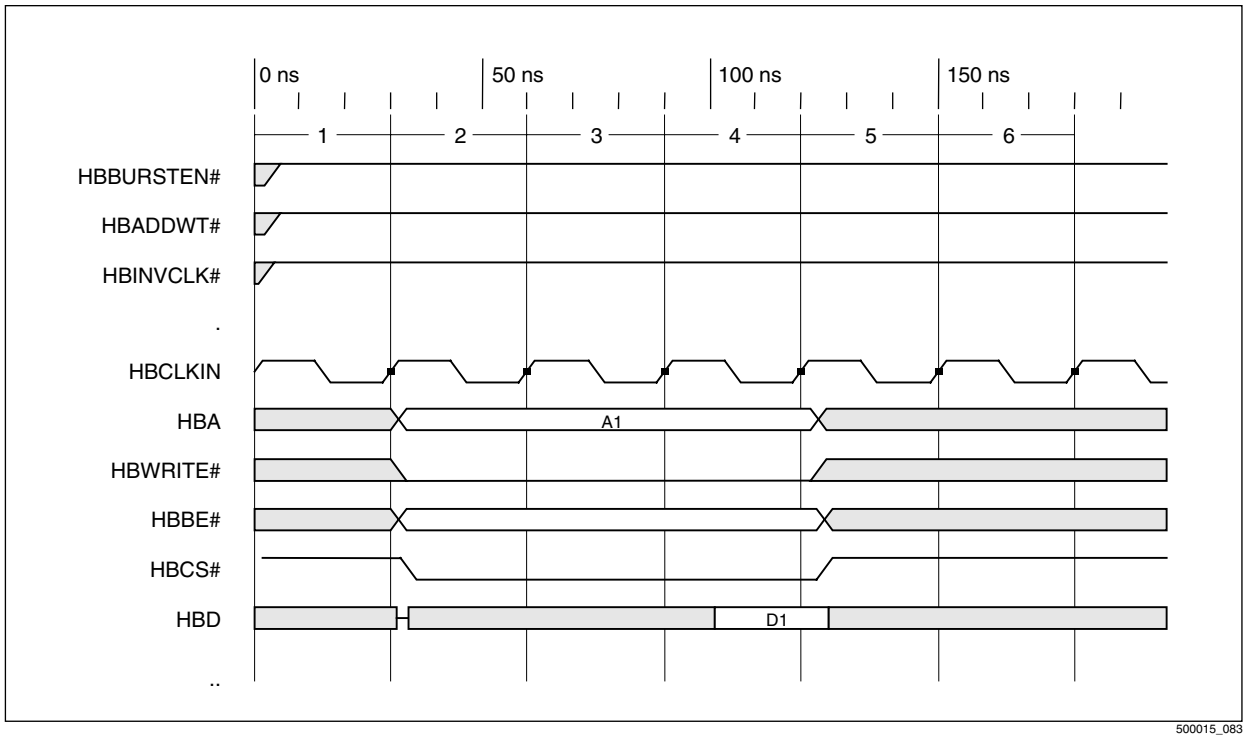


Figure 5-7. Host Bus Non-burst Read

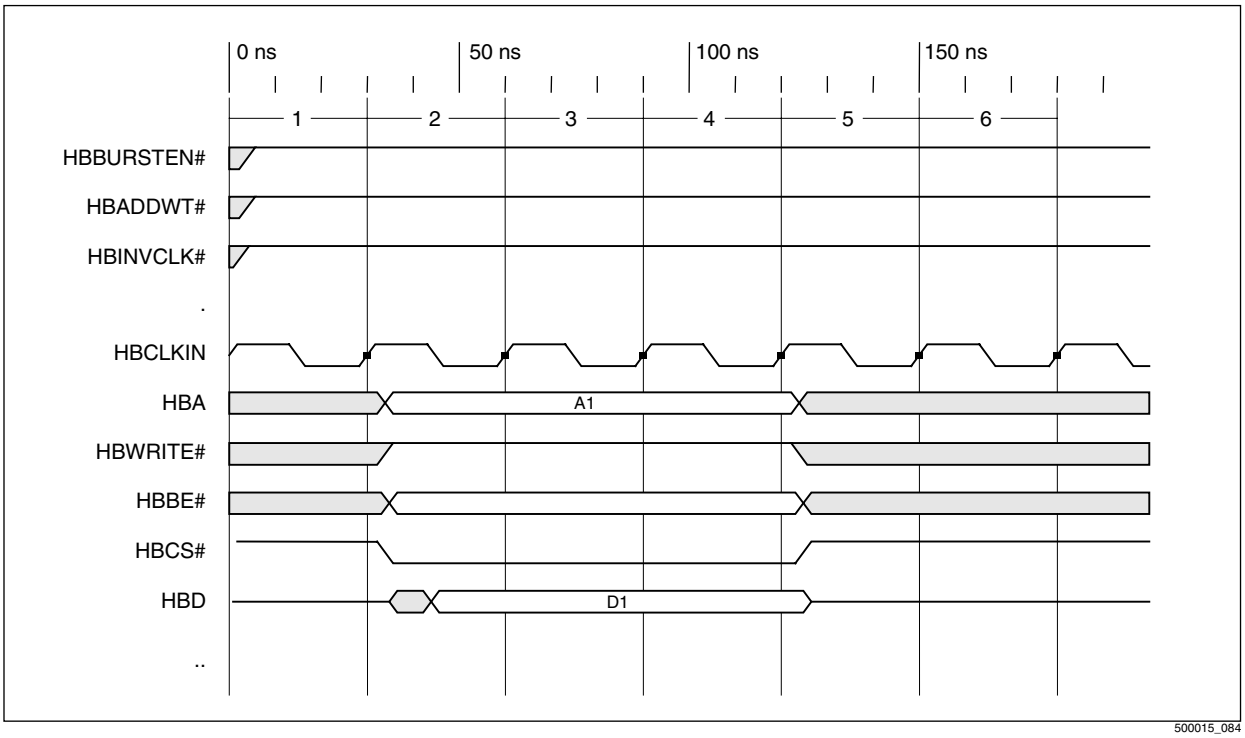


Figure 5-8. Host Bus Burst Read with Wait State

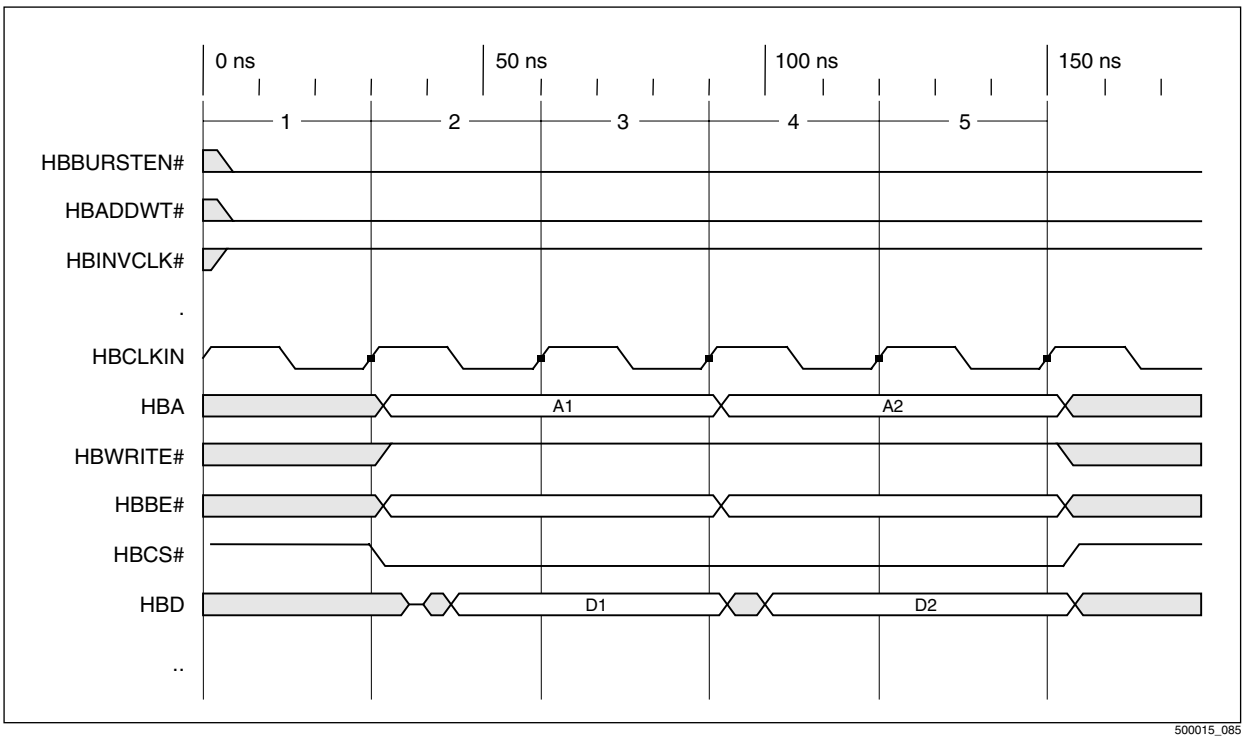


Figure 5-9. Host Bus Burst Write with Wait State

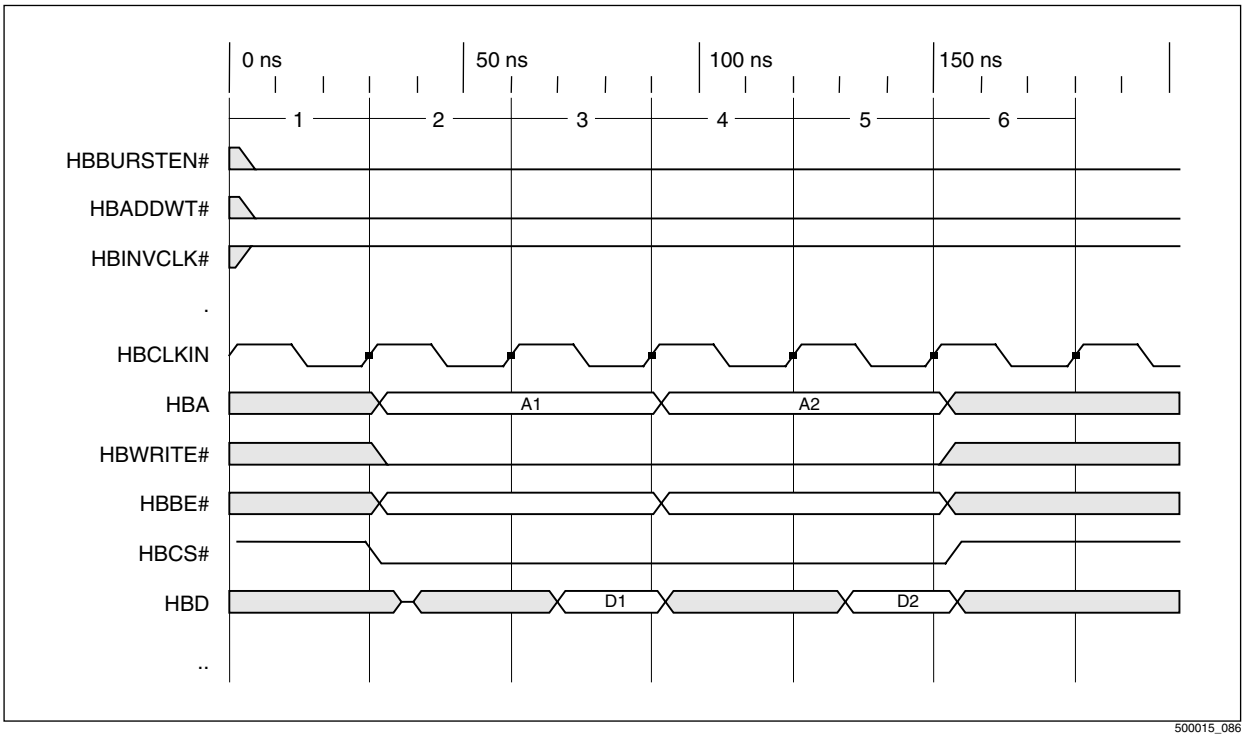


Figure 5-10. Host Bus Burst Write with Clock Inverted

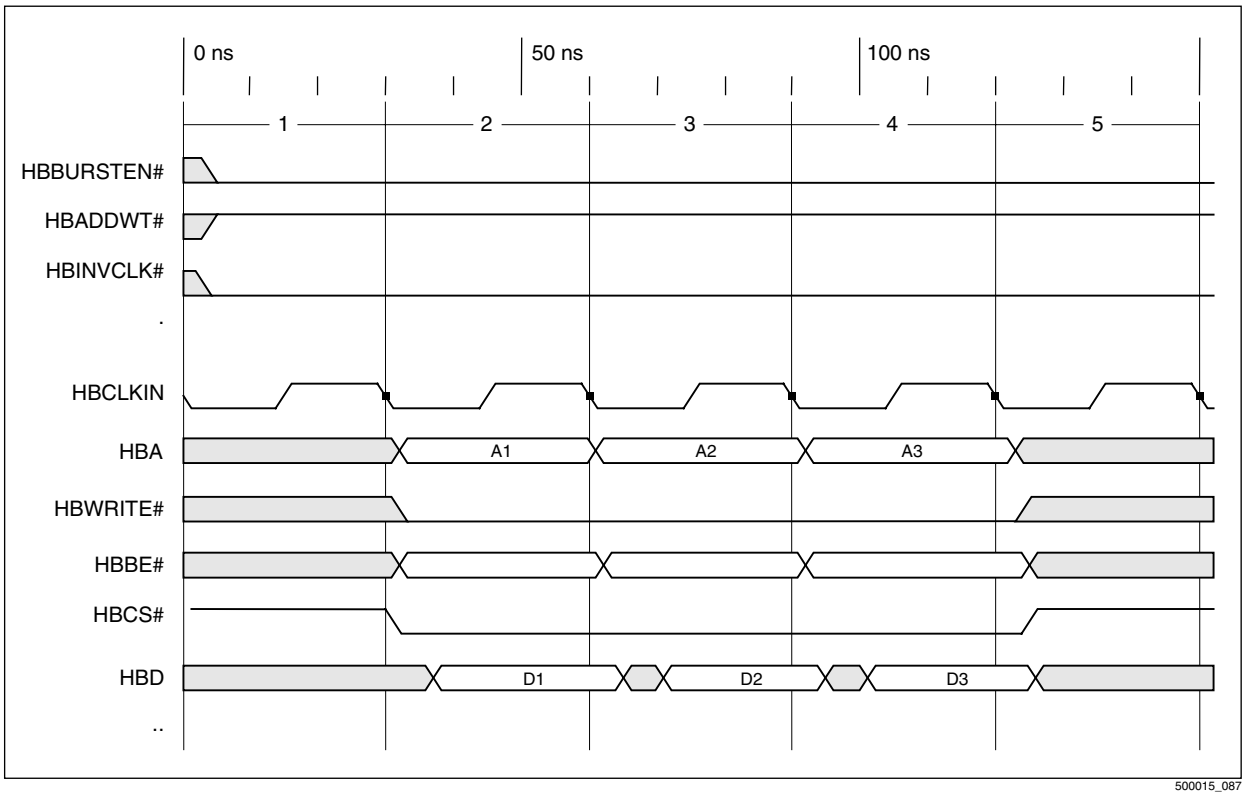
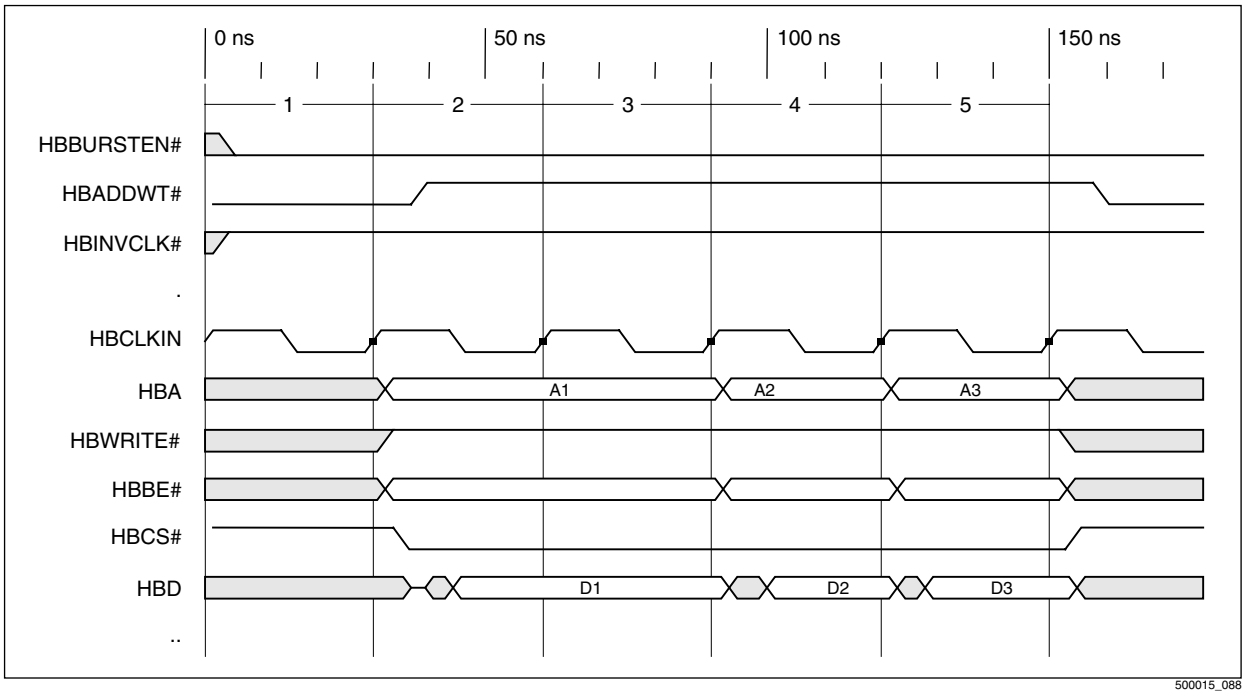


Figure 5-11. Host Bus Burst Read with Leading Wait State (HBADDWT# = not HBCS#)



5.2.5 FIFO Description

The FIFO block consists of two unidirectional first-in-first-out memories; TX FIFO and RX FIFO. The flow of data in TX FIFO is from the host bus to core APB, and the flow is opposite in RX FIFO.

5.2.5.1 TX FIFO

TX FIFO uses four 64 x 8 RAM to form a 128-byte deep FIFO. Four 8-bit RAMs from a 32-bit wide bus access at core APB side. To transfer data through the TX FIFO, the host processor should write to the TX FIFO Data register. The data gets latched into a temporary Flip Flop then transferred to the FIFO RAM when the end of write cycle is detected. The host processor should make sure that the TX FIFO is not full before attempting a TX FIFO write operation. Writing to a full FIFO results in overwriting the temporary register but it will not be transferred to the FIFO and new data is lost.

Reading this register would result in reading the value stored in the temporary register and not the contents of the FIFO. If two bytes of data is written to TX FIFO simultaneously, HBD[7:0] takes precedence over HBD[15:8], and the data in HBD[7:0] will come out earlier then in HBD[15:8] at core APB if single byte is read.

5.2.5.2 RX FIFO

RX FIFO uses four 64 × 8 RAM to form a 128-byte deep FIFO. Four 8-bit RAMs from 32-bit wide bus access at core APB side. If two bytes are read simultaneously at the host bus from RX FIFO, the first byte written into the FIFO by core processor will map to HBD[7:0] and the next byte to HBD[15:8].

5.2.5.3 FIFO Interrupt

The host FIFO status register contains the interrupt flags and related acknowledge bits.

When one interrupt flag is set, the HBINT# pin will be active if the corresponding interrupt enable bit is set to 1. To clear the interrupt two steps must be taken:

1. Service the interrupt.
2. Acknowledge the interrupt by writing a 1 to the corresponding acknowledge bit.

Acknowledging the interrupt will not clear it if the cause of the interrupt still exists. For example, if TXE interrupt is set because the TX FIFO is empty and we write 1 to bit 0 of the interrupt acknowledge register to clear the interrupt without writing anything to the TXFIFO, the interrupt bit will not be cleared.

All interrupts could be acknowledged simultaneously. Writing a 0 to the interrupt acknowledge bit will not affect anything.

What clears the interrupt is the action of writing to the register, not the content of it. In other words, if 1 is written to an acknowledge bit and the interrupt was cleared, the content of the register is not self-clearing. So, if the same interrupt takes place later, and even if the content of the corresponding acknowledge bit is still set to 1, a 1 should be rewritten to that bit to clear the new interrupt.

5.2.6 MAIL BOX Description

The MAILBOX consists of eight registers that can be read or written by either the host or the core processor. The first four registers are called TX MAIL[0 to 3], and the last four registers are RX MAIL[0 to 3]. In normal circumstances, the TX MAIL registers are written by the host processor and read by the core processor, while RX MAIL registers are written by the core processor and read by the host processor. TX MAIL3 has a special feature that allows the host processor to send an interrupt to the core processor by writing to this register. The core processor then acknowledges the interrupt by writing to APB FIFO Interrupt Acknowledge register bit 2 to clear the interrupt and, at the same time, to send an interrupt to the host processor if enabled. Similarly, the core processor can send an interrupt to the host processor by writing to RX MAIL3, and the host processor to acknowledge it by writing to Host FIFO Interrupt Acknowledge register bit 6 to clear the interrupt and send an interrupt to the core processor if enabled.

5.2.7 Host Bus Accessible Registers

The following registers may be accessed on the Host Bus by the host processor application.

Table 5-5. Host Bus Accessible Registers

HBA[8:0]	Read	Write
0x100–0x1FF	Host RX FIFO Data	Host TX FIFO Data
0x000	Host FIFO Control	Host FIFO Control
0x004	Host FIFO Status	Host FIFO Interrupt Acknowledge
0x010	Host TX FIFO Size	Reserved
0x014	Host TX FIFO High Threshold	Host TX FIFO High Threshold
0x018	Host TX FIFO Low Threshold	Host TX FIFO Low Threshold
0x020	Host RX FIFO Size	Reserved
0x024	Host RX FIFO High Threshold	Host RX FIFO High Threshold
0x028	Host RX FIFO Low Threshold	Host RX FIFO Low Threshold
0x030	TX Mail0	TX Mail0
0x034	TX Mail1	TX Mail1
0x038	TX Mail2	TX Mail2
0x03C	TX Mail3	TX Mail3
0x040	RX Mail0	Reserved
0x044	RX Mail1	Reserved
0x048	RX Mail2	Reserved
0x04C	RX Mail3	Reserved
0x070	HBCFG	Host Bus Configuration Register

Table 5-6. Host TX FIFO Data—Write HBA[8:0] = 0x100–0x1FF

Field	Name	Description
0–15	Host TX FIFO Data	Transmitting FIFO data from host to device.

Table 5-7. Host RX FIFO Data—Read HBA = 0x100–0x1FF

Field	Name	Description
0–15	Host RX FIFO Data	Receiving FIFO data from device back to host.

Table 5-8. Host FIFO Control Register—Read/Write HBA = 0x000 {default 0x19}

Field	Name	Description
0	TXFF_EN	Enables TX FIFO operation. The FIFO is enabled on reset.
1	HBTXRQ_EN	Enables operation of HBREQW. Disabled on reset.
2	Reserved	—
3	RXFF_EN	Enables the RX FIFO operation. The FIFO is enabled on reset.
4	HBRXRQ_EN	Enables operation of HBREQR. Enabled on reset.
5	Reserved	—
6–7	Reserved	—
8	TXEIE	TX FIFO Empty Interrupt Enable.
9	TXTHIE	TX FIFO Threshold Interrupt Enable.
10	TXM3IE	TX MAIL3 Acknowledge Interrupt Enable.
11	Reserved	—
12	RXFIE	RX FIFO Full Interrupt Enable.
13	RXTHIE	RX FIFO Threshold Interrupt Enable.
14	RXM3IE	RX MAIL3 Interrupt Enable.
15	TRGABORTIE	Target Abort Interrupt Enable

Table 5-9. Host FIFO Interrupt Status Register—Read HBA = 0x004 {default 0x300}

Field	Name	Description
0–7	—	Reserved.
8	TXE	TX FIFO is empty.
9	TXTH	TX FIFO depth is less than the threshold value.
10	TXM3	The core processor has acknowledged the TXM3 FULL interrupt.
11	Reserved	—
12	RXF	The RX FIFO is full.
13	RXTH	The RX_FIFO depth is greater than the threshold value.
14	RXM3	The core processor has written data to the RXMAIL3 register.
15	TRGABORT	Not applicable.

Table 5-10. Host FIFO Interrupt Acknowledge Register—Write HBA = 0x004

Field	Name	Description
0	TXEIAK	Writing a 1 Clears the corresponding interrupt status bit.
1	TXTHIAK	Writing a 1 Clears the corresponding interrupt status bit.
2	TXM3IAK	Writing a “1” Clears the corresponding interrupt status bit.
3	Reserved	—
4	RXFAK	Writing a 1 Clears the corresponding interrupt status bit.
5	RXTHAK	Writing a 1 Clears the corresponding interrupt status bit.
6	RXM3IAK	Writing a 1Clears the corresponding interrupt status bit.
7	Reserved	—
8	HSRESET	Host Soft Reset. Writing a 1 resets the entire device. The action is equivalent to resetting the device via the power-on reset pin.
9	TRGABORTIAK	Writing a 1 Clears the corresponding interrupt status bit.
10–15	Reserved	—

Table 5-11. Host TX FIFO Size—Read HBA = 0x010 {default 0x0}

Field	Name	Description
0–15	Host TX FIFO Size	The current depth of the transmitting FIFO.

Table 5-12. Host TX FIFO High Threshold—Read/Write HBA = 0x014 {default 0xF3}

Field	Name	Description
0–15	Host TX FIFO High Threshold	The high threshold value of the TX FIFO. The high threshold value of the TX FIFO. A TX FIFO depth greater than this value causes an interrupt to the host processor if enabled. Reading returns the last value written. Max value 0xF3.

Table 5-13. Host TX FIFO Low Threshold—Read/Write HBA = 0x018 {default 0x40}

Field	Name	Description
0–15	Host TX FIFO Low Threshold	The low threshold value of the TX FIFO. A TX FIFO depth less than this value causes an interrupt to the host if enabled. Reading returns the last value written.

Table 5-14. Host RX FIFO Size—Read HBA = 0x020 {default 0x0}

Field	Name	Description
0–15	Host RX FIFO Size	The current depth of the receiving FIFO.

Table 5-15. Host RX FIFO High Threshold—Read/Write HBA = 0x024 {default 0xC0}

Field	Name	Description
0–15	Host RX FIFO High Threshold	The high threshold value of the RX FIFO. Exceeding this value causes an interrupt to the host if enabled. The interrupt cannot be cleared until sufficient data has been removed from the FIFO by the host to cause the size to be less than the threshold. Reading returns the last value written.

Table 5-16. Host RX FIFO Low Threshold—Read/Write HBA = 0x028 {default 0x0D}

Field	Name	Description
0–15	Host RX FIFO Low Threshold	The low threshold value of the RX FIFO. A RX FIFO depth below this value causes an interrupt to the host processor if enabled. Reading returns the last value written. Minimum value is 0x0D.

Table 5-17. HBCFG—Read/Write HBA = 0x070 {default 0x00}

Field	Name	Description
0	8 Bit Host Bus Interface Mode	0 = 16 bit mode: both HBD[15:8] and HBD[7:0] are used to access the internal registers and FIFO. 1 = 8 bit mode: Only HBD[7:0] is used to access the internal register and FIFO.
1–7	Reserved	—

5.3 Expansion Bus Interface

This section describes the expansion bus for the CX28985 which implements the interface between the core processor and external memory.

5.3.1 SDRAM Interface

The SDRAM interface controls data flow to/from SDRAM memory in burst mode and is capable of bursting data in words, half words and byte sizes. It supports up to two memories, and is programmable to interface with 16 or 32 bits, 2 or 4 banks SDRAMs. The maximum memory space that can be covered is 16 MB on each chip select (CS). The SDRAM interface meets the requirements of most memory manufacturers, through the use of programmable timing parameters, and non-specific general commands.

5.3.2 Typical SDRAMs

The CX28985 provides address, data, and control lines for connection to a minimum of 64 Mb (2 MB × 32 bits) Synchronous Dynamic RAM (SDRAM) meeting the Intel PC100 Specification. This may be a single 64 Mb (1 × 2 M × 32) SDRAM which uses EXP_CS0# for chip select. A second 64 Mb (2 MB × 32 bits) SDRAM may be connected using EXP_CS1# for chip select. Alternatively, two 64 Mb (2 × 4 MB × 16 bits) SDRAMs can be connected using CS0 for chip select to both SDRAMs.

Typical SDRAMs that meet CX28985 requirements are listed in Table 5-18.

Table 5-18. Typical SDRAMs +3.3 V 64-Mb (2 M × 32 Bits Wide)

Manufacturer	Part Number
Toshiba	TC59S6432CFT/CFT-80
Samsung	K4S643232C-TC/L80
Micron	MT48LC2M32B2TG-8

5.4 Zipwiremulti Multiplexed ATM and PCM Signals

5.4.1 Overview

ZipWireMulti supports six main operational modes, where the mode determines which signal is assigned to a pin. See Figure 6-2.

These six modes are:

1. Framer bypass mode: Data bypasses the framer and ATM-TC entirely and interfaces with each of the DSP's individual raw data streams (x8).
2. Per-channel time slots G.shdsl transport mode: Data comes in on the individual PCM channels, framed up by each DSL framer (using G.shdsl frame format) and is transported on the DSL channel.
3. PCM highway time slots G.shdsl transport mode: A single-speed PCM input containing up to 288 time slots (18,432 Kbps) is connected to ZipWireMulti PCM interface. Each individual channel's framer can select up to 36 time slots from the highway for G.shdsl framing and transmission.
4. ATM transport, clear channel mode: ATM data is provided via the Utopia Level 2 interface to the ATM-TC block and then directly to the DSP (no G.shdsl framing is done).
5. ATM G.shdsl transport (aligned/Unaligned): ATM data is provided via the Utopia Level 2 interface to the ATM-TC block, then framed up by each DSL framer (using G.shdsl frame format) and transported on the DSL channel.
6. Simultaneous Transport of ATM and PCM highway: Mode 3 and mode 5 simultaneously.

NOTE: All 8 channels are programmed to the same mode (1 to 6), but each framer can then be independently programmed to the desired configuration of each specific channel (Frame format, time slots to be mapped, PCM or ATM data source per time slot basis).

5.4.2 Simultaneous Transport of ATM + PCM highway (Mode 6)

Figure 5-12 illustrates a detailed block diagram of the ZipWireMulti Transceiver/Framer and the data path when supporting simultaneous transport of ATM and PCM highway (mode 6).

In this mode, a standard Utopia Level 2 interface and a standard Concentration PCM Highway Interface (see description below) can be simultaneously mapped into the xDSL framer. Each xDSL framer is fully programmable and each xDSL channel can be independently programmed to map either to ATM or PCM per time slot basis. For example, when an xDSL channel payload contains M time slots, then N time slots can be mapped to ATM and (M–N) time slots can then be mapped to PCM. ($0 \le N \le M$).

Figure 5-12. CX28985 ATM + PCM highway Block Diagram

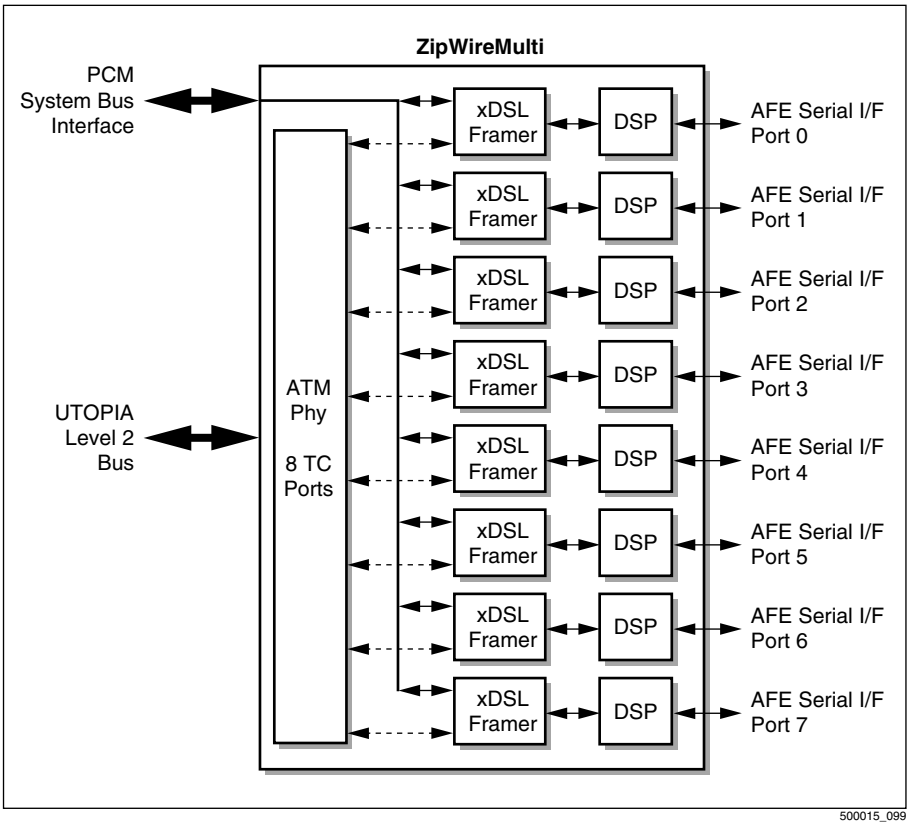
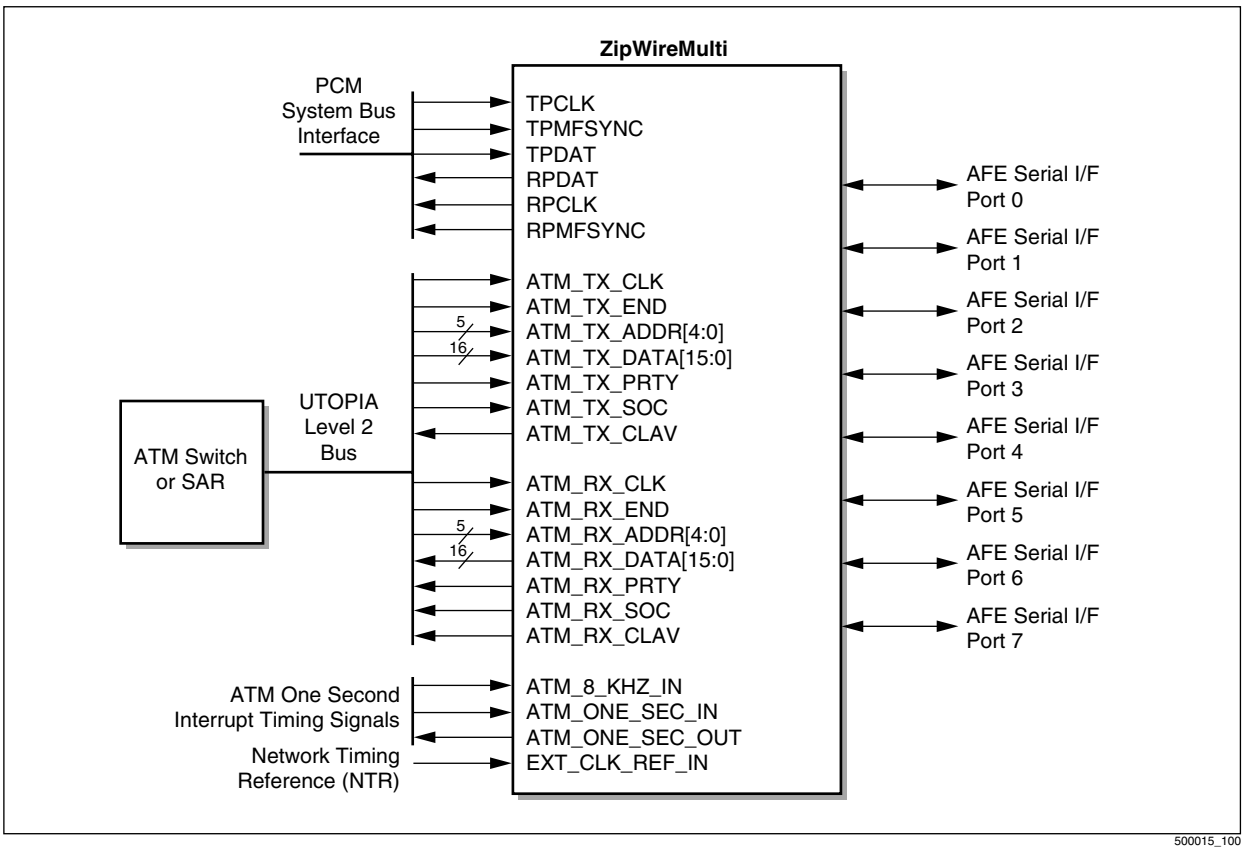


Figure 5-13 illustrates the ZipWireMulti Utopia Level 2 and PCM interface pins.

Figure 5-13. CX28985 ATM and PCM Interface



5.4.3 PCM System Bus Interface

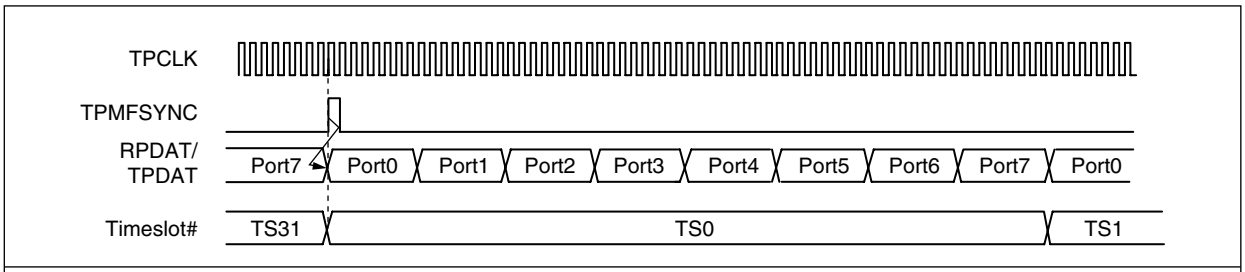
The PCM system bus interface is compatible with AT&T CHI interface, MITEL ST-BUS interface and can run up to 18,432 kHz (8 × 36 × 64 KHz = 288 time slots).

In this example, each time slot is mapped to a different port in Interleaving process, TS0 is mapped to all 8 ports, then TS1,TS2...TS31.

Each PCM time slot can be mapped to any DSL channel to satisfy different applications.

Figure 5-14 illustrates the PCM system bus timing when running at 8E1 rate.

Figure 5-14. System Bus timing in 16.384 MHz (8E1) using Time Slot Interleaving Mode



NOTE(S):
1. ZipWireMulti can operate as a master to the PCM Bus.
2. In the Master mode RPCLK and RPMFSYNC output signals are used.

500015_101

5.4.4 Utopia Interface and ATM-TC

The ZipWireMulti integrated 8-port ATM-TC provides a complete TC sublayer support as specified in ATM Forum af-phy-0043.000. Each TC port can be independently configured to operate at speeds ranging from 64 Kbps to 16 Mbps.

Up to four ZipWireMulti can be connected to the same Utopia Level 2 bus and form only four electrical loads for 32 ports, instead of 32/16 loads for single or dual transceiver. (The Utopia standard assumes a maximum of eight electrical loads).

The ATM-TC layer is compatible with ITU-T Recommendation I.432 and provides the following functions:

- Rate decoupling between the ATM layer and the synchronous (or plesiochronous) G.shdsl.
- Insertion of IDLE cells
- Insertion of ATM Header Error Check (HEC) byte
- Cell payload scrambling and descrambling
- Octet and bit-level cell delineation in the receive channel

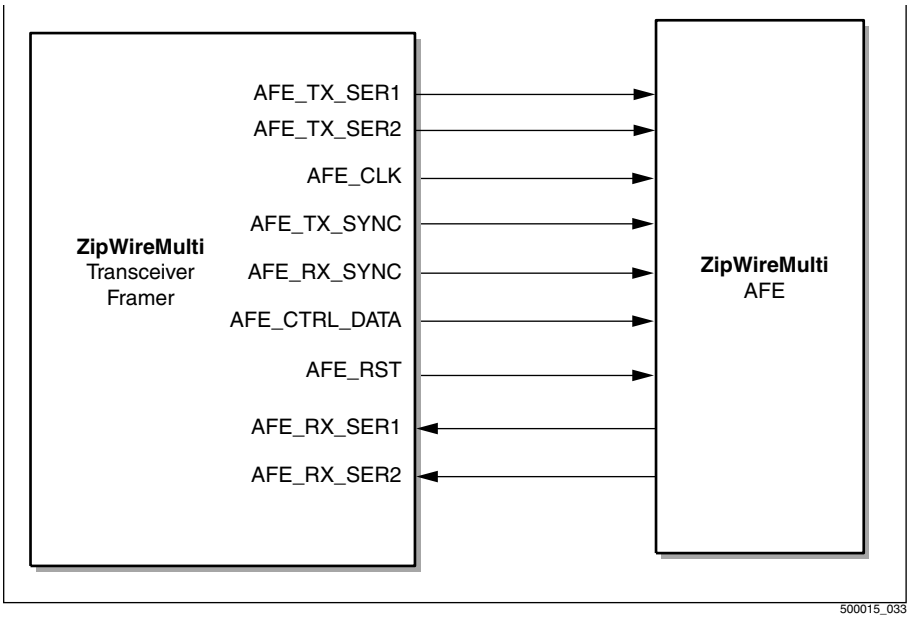
The Utopia interface is compatible with Utopia Level 2 as described in ATM Forum af-phy-0039.000:

- 8/16-bit data path interface
- Multi-PHY capability
- 50 MHz maximum clock rate

5.5 ZipWireMulti Transceiver/Framer to AFE Interface

Figure 5-19 illustrates the ZipWireMulti Transceiver/Framer to AFE interface. The two devices must be connected as shown.

Figure 5-15. ZipWireMulti Transceiver/Framer to AFE Interface

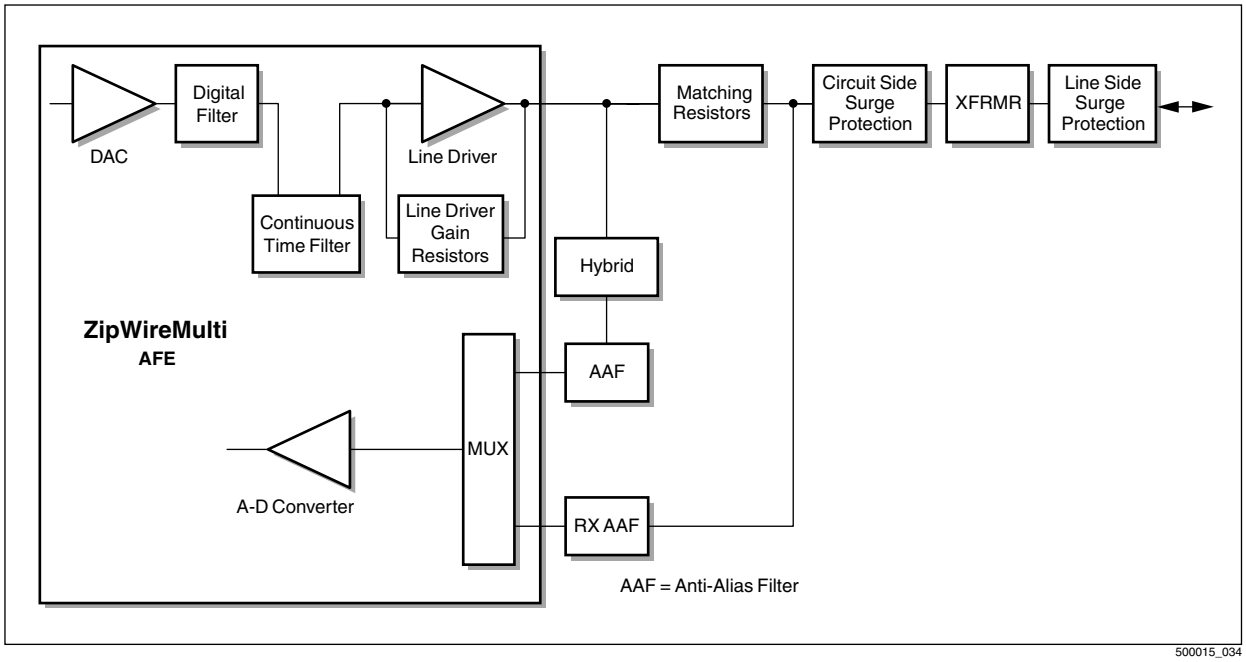


5.6 Transmission Line Interface

The DSL interface consists of the continuous time filter, line drive feedback resistors, impedance matching resistors, compromise hybrid, transformer, and surge protection. All signals are differential pairs. Only NPO-type capacitors should be used in the DSL transmission line interface except for the surge protection blocks. The NPO capacitors are selected because of their high linearity characteristics. All capacitors should be 5 percent tolerant while the resistors should be 1 percent tolerant.

Figure 5-20 illustrates a block diagram of the DSL transmission line interface.

Figure 5-16. DSL Transmission Line Interface

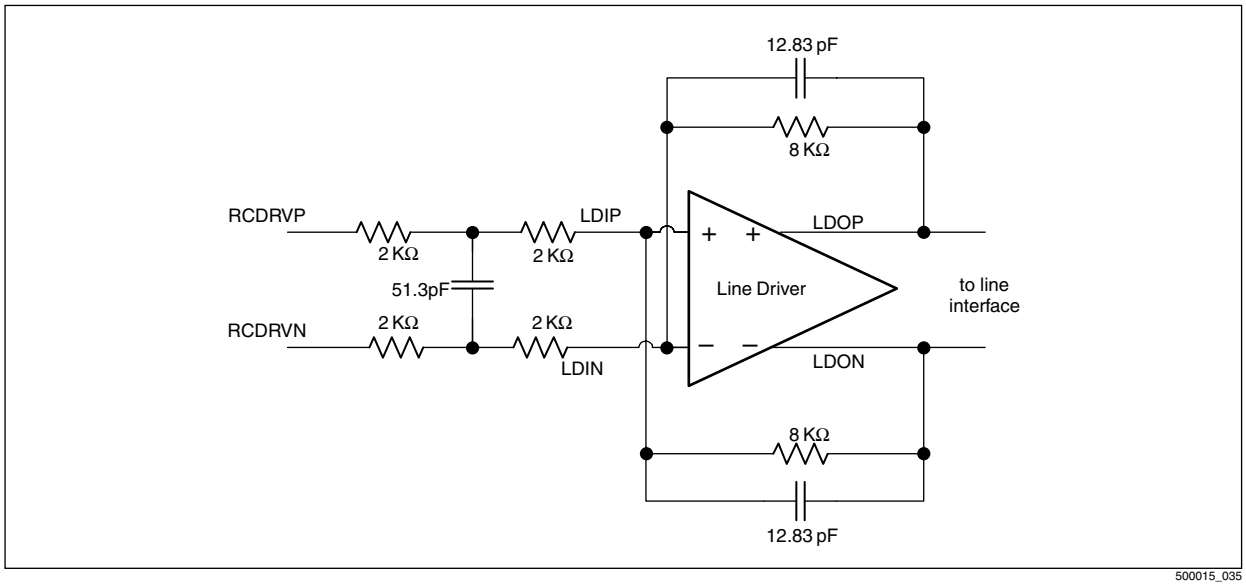


5.6.1 Continuous Time Filter and Line Driver Control

The internal Continuous Timer Filter filters out clock images created from the switched-capacitor filters. The internal line driver gain control resistors set the line driver gain.

Figure 5-17 illustrates the internal Continuous Time Filter and Line Driver Control connections.

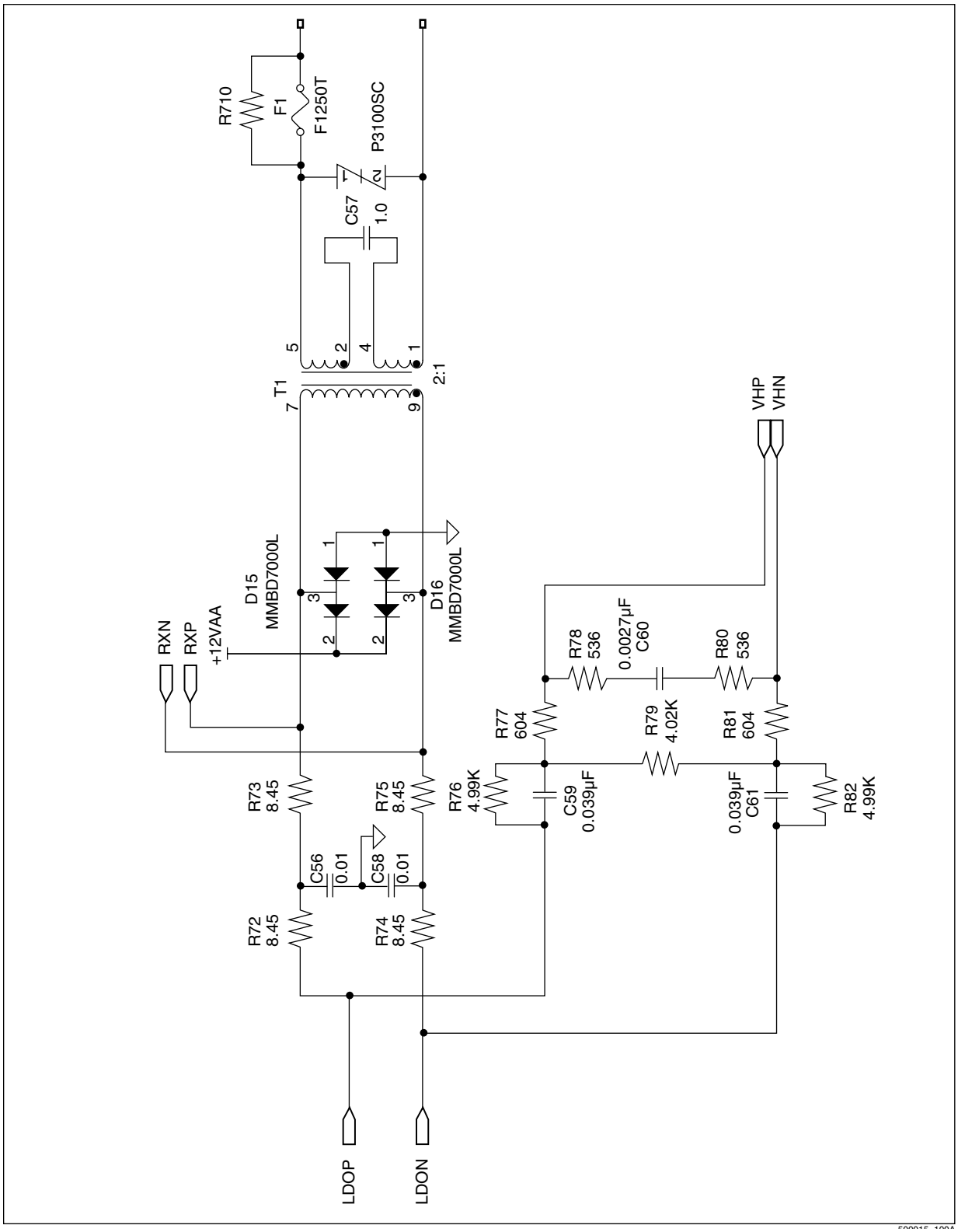
Figure 5-17. Continuous Time Filter and Line Driver Control



5.6.2 Compromise Hybrid, Matching Resistors, and Transformer

Figure 5-18 illustrates the hybrid topology.

Figure 5-18. Hybrid Topology



5.6.2.1 Compromise Hybrid

The purpose of the compromise hybrid is to model the impedance of the transmission line. This model generates an approximation of the transmitted signal's echo. The echo replica is then subtracted from the signal on the line transformer to generate a first-order approximation of the received signal. Although the CX28985 contains a digital Echo Canceller (EC), the hybrid is needed to reduce the signal-level input to the Analog-to-Digital Converter (ADC). This eliminates ADC overflow on short loops and increases the resolution of the digitized received signal for better digital signal processing performance.

In the hybrid section, two capacitors are placed in parallel to achieve nonstandard capacitor values.

5.6.2.2 Impedance Matching Resistors

Impedance-matching resistors (2.49 Ω) are placed in the transmit path so the output impedance of the line interface more closely matches the impedance of the transmission line and load. This maximizes the power transferred to the receiver on the other end of the line. The load is assumed to be 135 Ω.

5.6.2.3 Transformer

The line transformer provides DC isolation from the transmission line by creating a high-pass filter. The winding ratio of the transformer must be 5.0:1 (line side:circuit side) to generate the appropriate voltage level on the line. The primary inductance (L) of the transformer (line side) is a very critical parameter. If the inductance is too high, the cutoff frequency of the filter will be too low, and the CX28985 Echo Canceller and Equalizer will not be able to cancel out the low frequency components of the echo and Inter-Symbol Interference (ISI). If L is too low, part of the information in the signal will be filtered out, thereby decreasing the Signal-to-Noise (SNR) ratio. In addition, the line transformer must meet certain return loss requirements to maximize system performance.

5.6.2.4 Anti-Alias Filters

Anti-aliasing filters are needed to filter out high frequencies that would be aliased back into the passband as noise. These filters are made of all passive components. The cutoff frequency (f_c) is designed to be as low as possible to achieve maximum attenuation of aliasing frequencies without filtering out the desired signal.

5.6.3 Surge Protection

TBD

5.7 Voltage Reference and Compensation Circuitry

Compensation capacitors must be connected between all CX28985 voltage reference pins and analog ground. The voltage reference signals, their associated pin numbers, and the recommended compensation capacitor values are listed in Table 5-19.

Table 5-19. ZipWireMulti AFE Compensation Capacitor Values

Signal Name	Value
VRNTX	0.1 µF
VRPTX	0.1 µF
VCMI	0.1 µF
VCMO	0.1 µF
VBGN	0.1 µF
VBGP	0.1 µF
VRNRX	0.1 µF
VRPRX	0.1 µF

In addition to the compensation capacitors, external passive components are needed to set the bias current used in the CX28985. This network is illustrated in Figure 5-19. The recommended value of the resistors are listed in Table 5-20.

Figure 5-19. ZipWireMulti AFE Bias Current Network

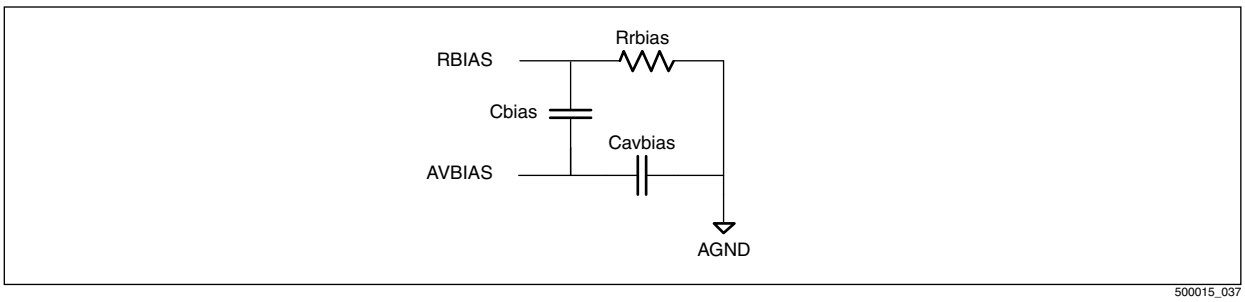


Table 5-20. ZipWireMulti AFE Bias Current Network Values

Signal Name	Value
Rrbias	10.0 kΩ
Cbias	0.1 µF
Cavbias	0.1 µF

5.8 Test and Diagnostic Interface (JTAG)

The Test and Diagnostic interface comprises a test access port and two Serial Test Ports (STP). The test access port conforms to *IEEE Std. 1149.1-1990 (IEEE Standard Test Access Port and Boundary Scan Architecture)*. Also referred to as Joint Test Action Group (JTAG), this interface provides direct serial access to each of the transceiver's I/O pins. This capability can be used during an in-circuit board test to increase the testability and reduce the cost of the in-circuit test process.

The serial test ports can be viewed as a real-time virtual probe for looking at the transceiver's internal signals. A majority of the receiver's signal path is accessible through these outputs.

6.0 Pin Descriptions

The ZipWireMulti solution is available in a 9-device chip set consisting of a ZipWireMulti Transceiver/Framer and 8 ZipWireMulti AFE/Line Drivers.

6.1 ZipWireMulti Pin Assignments

This section provides the pin assignments for the ZipWireMulti devices.

6.1.1 CX28985 DSP/Framer Single Mode Ball Assignment

Table 6-2 lists the CX28985 DSP/framer multi mode ball assignments. Section 5.4 describes how the signals on these balls vary, depending on the selected mode.

Table 6-1. CX28985 DSP/Framer Single Mode Ball Assignment (1 of 2)

Ball	Signal Name
A02	EXP_ADDR(00)
A03	EXP_CS(5)
A04	EXP_CS(3)
A05	EXP_CS(0)
A06	TDO
A07	TCK
A08	UART_RX
A09	GPIO(5)
A10	GPIO(2)
A11	XTALO_B
A12	XTALI
A13	HBBE0#
A14	HBA(8)
A15	HBD(13)
A16	HBWRITE#
A17	HP_ADD_WT
A18	HBD(09)
A20	HBA(5)
A21	HBD(04)
A22	HBA(2)
A23	HBD(00)
A24	TEST_TM(0)
A25	3_STATE#
A26	TEST_CLK
B04	EXP_CS(4)
B05	EXP_CS(1)
B06	TMS
B07	TRST#
B08	GPIO(7)
B09	GPIO(4)
B10	GPIO(1)
B11	HPBURSTEN#
B12	HP_INVCLK
B13	HBD(15)
B14	HBA(7)
B15	HBD(12)
B16	HPCS#
B17	HBINT#
B18	HBA(6)
B19	HBD(07)
B20	HBA(4)
B21	HBD(03)
B22	HBD(02)
B23	TEST_TM(2)
B24	TEST_ST#
B25	TEST_SM
B26	TEST_SE(2)
C01	EXP_ADDR(01)
C05	EXP_CS(2)
C06	TDI
C07	UART_TX
C08	GPIO(6)
C09	GPIO(3)
C10	GPIO(0)
C11	XTALO
C12	HBBE1#
C13	HBD(14)
C15	HBD(11)
C16	HBCLKIN
C17	HBD(10)
C18	HBD(08)
C19	HBD(06)
C20	HBD(05)
C21	HBA(3)
C22	HBD(01)
C23	TEST_TM(1)
C24	TEST_SE(1)
C25	TEST_SE(0)
C26	RST#
D01	EXP_DATA(01)
D02	EXP_DATA(00)
D24	PLLBYPASS
D25	EXP_EXT_BOOT#
E01	EXP_ADDR(03)
E02	EXP_ADDR(02)
E03	EXP_DATA(02)
F01	EXP_DATA(05)
F02	EXP_DATA(04)
F03	EXP_DATA(03)
G01	EXP_ADDR(05)
G02	EXP_DATA(06)
G03	EXP_ADDR(04)
H01	EXP_DATA(08)
H02	EXP_DATA(07)
H03	EXP_ADDR(06)
J01	EXP_DATA(10)
J02	EXP_ADDR(07)
J03	EXP_DATA(09)
K01	EXP_DQM(0)
K02	EXP_CAS#
K03	EXP_DATA(11)
L01	EXP_DQM(3)
L02	EXP_DQM(2)
L03	EXP_DQM(1)
M01	EXP_WE#
M02	EXP_RD#

Table 6-1. CX28985 Single Mode Ball Assignment (2 of 2)

Ball	Signal Name
M03	EXP_RAS#
N01	EXP_ADDR(08)
N02	EXP_CLK
N03	EXP_CLKE
P01	EXP_DATA(13)
P02	EXP_DATA(12)
P03	EXP_ADDR(09)
R01	EXP_DATA(15)
R02	EXP_ADDR(10)
R03	EXP_DATA(14)
T01	EXT_CLK_REF_OUT
T02	EXP_ADDR(11)
T03	EXP_DATA(16)
U01	EXP_DATA(18)
U02	EXP_DATA(17)
U03	EXP_ADDR(12)
V01	EXP_ADDR(14)
V02	EXP_DATA(19)
V03	EXP_ADDR(13)
W01	EXP_DATA(20)
W02	EXP_DATA(21)
W03	EXP_DATA(22)
Y01	EXP_ADDR(15)
Y02	EXP_DATA(23)
Y03	EXP_DATA(24)
AA01	EXP_ADDR(16)
AA02	EXP_ADDR(17)
AA03	EXP_DATA(25)
AB01	EXP_DATA(26)
AB02	EXP_DATA(27)
AB03	EXP_ADDR(18)
AC01	EXP_ADDR(19)
AC02	EXP_DATA(28)
AC03	EXP_DATA(29)
AD01	EXT_CLK_REF_SEL(1)
AD02	EXT_CLK_REF_SEL(2)

Ball	Signal Name
AD03	AFE_CLK(0)
AD04	AFE_TX_SER2(0)
AD05	AFE_RX_SER2(1)
AD06	AFE_RX_SYNC(1)
AD07	AFE_TX_SYNC(1)
AD08	AFE_CTRL_DATA(2)
AD09	AFE_TX_SER1(2)
AD10	AFE_RX_SER1(3)
AD11	AFE_CLK(3)
AD12	AFE_TX_SER2(3)
AD13	EXT_CLK_REF_IN
AD14	AFE_CTRL_DATA(4)
AD15	AFE_TX_SER1(4)
AD16	AFE_RX_SER1(5)
AD17	AFE_CLK(5)
AD18	AFE_TX_SER2(5)
AD19	AFE_RX_SER2(6)
AD20	AFE_RX_SYNC(6)
AD21	AFE_TX_SYNC(6)
AD22	AFE_CTRL_DATA(7)
AD23	AFE_TX_SER1(7)
AE01	EXT_CLK_REF_SEL(0)
AE02	AFE_RX_SER1(0)
AE03	AFE_CTRL_DATA(0)
AE04	AFE_TX_SER1(0)
AE05	AFE_RX_SER1(1)
AE06	AFE_CLK(1)
AE07	AFE_TX_SER2(1)
AE08	AFE_RX_SER2(2)
AE09	AFE_RX_SYNC(2)
AE10	AFE_TX_SYNC(2)
AE11	AFE_CTRL_DATA(3)
AE12	AFE_TX_SER1(3)
AE13	AFE_RST
AE14	AFE_RX_SER2(4)
AE15	AFE_RX_SYNC(4)

Ball	Signal Name
AE16	AFE_TX_SYNC(4)
AE17	AFE_CTRL_DATA(5)
AE18	AFE_TX_SER1(5)
AE19	AFE_RX_SER1(6)
AE20	AFE_CLK(6)
AE21	AFE_TX_SER2(6)
AE22	AFE_RX_SER2(7)
AE23	AFE_RX_SYNC(7)
AE24	AFE_TX_SYNC(7)
AF01	EXP_DATA(30)
AF02	EXP_DATA(31)
AF03	AFE_RX_SER2(0)
AF04	AFE_RX_SYNC(0)
AF05	AFE_TX_SYNC(0)
AF06	AFE_CTRL_DATA(1)
AF07	AFE_TX_SER1(1)
AF08	AFE_RX_SER1(2)
AF09	AFE_CLK(2)
AF10	AFE_TX_SER2(2)
AF11	AFE_RX_SER2(3)
AF12	AFE_RX_SYNC(3)
AF13	AFE_TX_SYNC(3)
AF14	AFE_RX_SER1(4)
AF15	AFE_CLK(4)
AF16	AFE_TX_SER2(4)
AF17	AFE_RX_SER2(5)
AF18	AFE_RX_SYNC(5)
AF19	AFE_TX_SYNC(5)
AF20	AFE_CTRL_DATA(6)
AF21	AFE_TX_SER1(6)
AF22	AFE_RX_SER1(7)
AF23	AFE_CLK(7)
AF24	AFE_TX_SER2(7)

Table 6-2. CX28985 DSP/Framer Multi-Mode Ball Assignments (1 of 2)

Ball	Mode 1	Mode 2	Mode 4 and 5	Mode 3 and 6
D26	N/C (1)	N/C (1)	ATM_TX_SOC	ATM_TX_SOC
E24	N/C (1)	N/C (1)	ATM_TX_ENB#	ATM_TX_ENB#
E25	N/C (1)	N/C (1)	ATM_TX_PRTY	ATM_TX_PRTY
E26	HXP(7)	RPMFSYNC(7)	ATM_TX_CLAV	ATM_TX_CLAV
F24	N/C (1)	TPCLK(6)	ATM_TX_ADDR(4)	ATM_TX_ADDR(4)
F25	N/C (1)	TPCLK(5)	ATM_TX_ADDR(3)	ATM_TX_ADDR(3)
F26	N/C (1)	TPCLK(4)	ATM_TX_ADDR(2)	ATM_TX_ADDR(2)
G24	N/C (1)	TPCLK(3)	ATM_TX_ADDR(1)	ATM_TX_ADDR(1)
G25	N/C (1)	TPCLK(2)	ATM_TX_ADDR(0)	ATM_TX_ADDR(0)
G26	HXP(6)	RPMFSYNC(6)	ATM_RX_SOC	ATM_RX_SOC
H24	HXP(5)	RPMFSYNC(5)	ATM_RX_PRTY	ATM_RX_PRTY
H25	N/C (1)	TPCLK(1)	ATM_RX_ENB#	ATM_RX_ENB#
H26	HXP(4)	RPMFSYNC(4)	ATM_RX_DATA(15)	ATM_RX_DATA(15)
J24	HXP(3)	RPMFSYNC(3)	ATM_RX_DATA(14)	ATM_RX_DATA(14)
J25	HXP(2)	RPMFSYNC(2)	ATM_RX_DATA(13)	ATM_RX_DATA(13)
J26	HXP(1)	RPMFSYNC(1)	ATM_RX_DATA(12)	ATM_RX_DATA(12)
K24	RXDAT(7)	RPDAT(7)	ATM_RX_DATA(11)	ATM_RX_DATA(11)
K25	RXDAT(6)	RPDAT(6)	ATM_RX_DATA(10)	ATM_RX_DATA(10)
K26	RXDAT(5)	RPDAT(5)	ATM_RX_DATA(09)	ATM_RX_DATA(09)
L24	RXDAT(4)	RPDAT(4)	ATM_RX_DATA(08)	ATM_RX_DATA(08)
L25	RXDAT(3)	RPDAT(3)	ATM_RX_DATA(07)	ATM_RX_DATA(07)
L26	N/C (1)	TPCLK(7)	ATM_TX_CLK	ATM_TX_CLK
M24	RXDAT(2)	RPDAT(2)	ATM_RX_DATA(06)	ATM_RX_DATA(06)
M25	RXDAT(1)	RPDAT(1)	ATM_RX_DATA(05)	ATM_RX_DATA(05)
M26	HXCLK(7)	RPCLK(7)	ATM_RX_DATA(04)	ATM_RX_DATA(04)
N24	HXCLK(6)	RPCLK(6)	ATM_RX_DATA(03)	ATM_RX_DATA(03)
N25	HXCLK(5)	RPCLK(5)	ATM_RX_DATA(02)	ATM_RX_DATA(02)
N26	HXCLK(4)	RPCLK(4)	ATM_RX_DATA(01)	ATM_RX_DATA(01)
P24	RPDPLLCLK	RPDPLLCLK	ATM_RX_CLK	ATM_RX_CLK
P25	HXCLK(2)	RPCLK(2)	ATM_RX_CLAV	ATM_RX_CLAV
P26	HXCLK(3)	RPCLK(3)	ATM_RX_DATA(00)	ATM_RX_DATA(00)
R24	N/C (1)	TPMFSYNC(5)	ATM_RX_ADDR(2)	ATM_RX_ADDR(2)
R25	N/C (1)	TPMFSYNC(6)	ATM_RX_ADDR(3)	ATM_RX_ADDR(3)
R26	N/C (1)	TPMFSYNC(7)	ATM_RX_ADDR(4)	ATM_RX_ADDR(4)
T24	N/C (1)	TPMFSYNC(0)	TPCLK(7)	TPMFSYNC(0)
T25	N/C (1)	TPMFSYNC(3)	ATM_RX_ADDR(0)	ATM_RX_ADDR(0)

Table 6-2. CX28985 DSP/Framer Multi-Mode Ball Assignments (2 of 2)

Ball	Mode 1	Mode 2	Mode 4 and 5	Mode 3 and 6
T26	N/C (1)	TPMFSYNC(4)	ATM_RX_ADDR(1)	ATM_RX_ADDR(1)
U24	HXP(0)	RPMFSYNC(0)	TPCLK(6)	RPMFSYNC(0)
U25	N/C (1)	TPCLK(0)	TPCLK(0)	TPCLK(0)
U26	TXDAT(0)	TPDAT(0)	TPCLK(2)	TPDAT(0)
V24	N/C (1)	PEXTCLK(0)	TPCLK(1)	PEXTCLK(0)
V25	HXCLK(0)	RPCLK(0)	TPCLK(4)	RPCLK(0)
V26	RXDAT(0)	RPDAT(0)	TPCLK(5)	RPDAT(0)
W24	N/C (1)	TPMFSYNC(2)	ATM_ONE_SEC_IN	ATM_ONE_SEC_IN
W25	HXCLK(1)	RPCLK(1)	ATM_ONE_SEC_OUT	OUT ATM_ONE_SEC_OUT
W26	N/C (1)	N/C (1)	TPCLK(3)	N/C(1)
Y24	N/C (1)	PEXTCLK(6)	ATM_TX_DATA(14)	ATM_TX_DATA(14)
Y25	N/C (1)	PEXTCLK(7)	ATM_TX_DATA(15)	ATM_TX_DATA(15)
Y26	N/C (1)	TPMFSYNC(1)	ATM_8_KHZ_IN	ATM_8_KHZ_IN
AA24	N/C (1)	PEXTCLK(3)	ATM_TX_DATA(11)	ATM_TX_DATA(11)
AA25	N/C (1)	PEXTCLK(4)	ATM_TX_DATA(12)	ATM_TX_DATA(12)
AA26	N/C (1)	PEXTCLK(5)	ATM_TX_DATA(13)	ATM_TX_DATA(13)
AB24	TXDAT(7)	TPDAT(7)	ATM_TX_DATA(07)	ATM_TX_DATA(07)
AB25	N/C (1)	PEXTCLK(1)	ATM_TX_DATA(09)	ATM_TX_DATA(09)
AB26	N/C (1)	PEXTCLK(2)	ATM_TX_DATA(10)	ATM_TX_DATA(10)
AC26	N/C (1)	N/C (1)	ATM_TX_DATA(08)	ATM_TX_DATA(08)
AD24	N/C (1)	N/C (1)	ATM_TX_DATA(00)	ATM_TX_DATA(00)
AD25	TXDAT(4)	TPDAT(4)	ATM_TX_DATA(04)	ATM_TX_DATA(04)
AD26	TXDAT(6)	TPDAT(6)	ATM_TX_DATA(06)	ATM_TX_DATA(06)
AE25	TXDAT(2)	TPDAT(2)	ATM_TX_DATA(02)	ATM_TX_DATA(02)
AE26	TXDAT(5)	TPDAT(5)	ATM_TX_DATA(05)	ATM_TX_DATA(05)
AF25	TXDAT(1)	TPDAT(1)	ATM_TX_DATA(01)	ATM_TX_DATA(01)
AF26	TXDAT(3)	TPDAT(3)	ATM_TX_DATA(03)	ATM_TX_DATA03)

NOTE(S):
1. N/C = No connect. Pins may be left open.
2. See Table 6-6 for signal descriptions and what to do if signals are not used.

Table 6-3. CX28985 DSP/Framer Power and Ground

Ball
VDD (VCORE +1.8 VDC)
E11, K4, K23, L4, L23, P4, P23, U4, U23, E5, E6, E7, E8, E9, E10, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E22, F4, F23, G4, G23, H4, H23, J4, J23, M4, M23, N4, N23, R4, R23, T4, T23, V4, V23, W4, W23, Y4, Y23, AA4, AA23, AB5, AB6, AB7, AB8, AB9, AB10, AB11, AB12, AB13, AB14, AB15, AB16, AB17, AB18, AB19, AB20, AB21, AB22
VDDO (VIO +3.3 VDC)
F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21, F22 G5, G22, H5, H22, J5, J22, K5, K22, L5, L22, M5, M22, N5, N22, P5, P22, R5, R22, T5, T22, U5, U22, V5, V22, W5, W22, Y5, Y22, AA5, AA6, AA7, AA8, AA9, AA10, AA11, AA12, AA13, AA14, AA15, AA16, AA17, AA18, AA19, AA20, AA21, AA22
VGNN (VESD)
E4, E23, AB4, AB23
GND
G6, G7, G8, G9, G10, G11, G12, G13, G14, G15, G16, G17, G18, G19, G20, G21, H6, H7, H8, H9, H10, H11, H12, H13, H14, H15, H16, H17, H18, H19, H20, H21, J6, J7, J8, J9, J10, J11, J12, J13, J14, J15, J16, J17, J18, J19, J20, J21, K6, K7, K8, K9, K10, K11, K12, K13, K14, K15, K16, K17, K18, K19, K20, K21, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19, L20, L21, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, M21, N6, N7, N8, N9, N10, N11, N12, N13, N14, N15, N16, N17, N18, N19, N20, N21, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20, P21, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, T17, T18, T19, T20, T21, U6, U7, U8, U9, U10, U11, U12, U13, U14, U15, U16, U17, U18, U19, U20, U21, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, W6, W7, W8, W9, W10, W11, W12, W13, W14, W15, W16, W17, W18, W19, W20, W21, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16, Y17, Y18, Y19, Y20, Y21

Table 6-4. CX28985 DSP/Framer No Connect Pins

Ball
Not Connected
A1, B1 B2, C2 B3, C3, D3 C4, D4, AC4 A19 C14 D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23 AC5, AC6, AC7, AC8, AC9, AC10, AC11, AC12, AC13, AC14, AC15, AC16, AC17, AC18, AC19, AC21, AC22, AC23, AC24, AC25

6.1.2 Pin List for ZipWireMulti Single Channel

Table 6-5. Pin List for 28985 Single Channel AFE/Line Drive—Alphabetic Order

Pin Number	Signal Name
3	A12GND
44	A12GND
11	A33GND
13	A33GND
35	AFE_CLK
31	AFE_CTRL_DATA
36	AFE_RST
26	AFE_RX_SER1
25	AFE_RX_SER2
27	AFE_RX_SYNC
29	AFE_TX_SER1
30	AFE_TX_SER2
28	AFE_TX_SYNC
10	AVBIAS
32	DGND
41	LD_MH
40	LD_ML
47	LD_PH
48	LD_PL
43	LDAON
45	LDAOP
1	LDON
2	LDOP
37	No Connect
38	No Connect
9	RBIAS
5	RCDRVN
6	RCDRVP
34	SCAN_EN
4	VA12
42	VA12
46	VA12
12	VA33
14	VA33
39	VA33
17	VBGN
18	VBGP
15	VCMI
16	VCMO
24	VHN
23	VHP
33	VD33
19	VRNRX
7	VRNTX
20	VRPRX
8	VRPTX
22	VXN
21	VXP

6.2 ZipWireMulti Signal Descriptions

This section provides the signal descriptions for both the ZipWireMulti Transceiver/Framer and ZipWireMulti AFE/Line Drive devices.

6.2.1 ZipWireMulti Transceiver/Framer Signal Descriptions

Table 6-6 lists the ZipWireMulti Transceiver/Framer Signal (pin) descriptions.

6.2.2 ZipWireMulti-Signal Descriptions

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (1 of 7)

Signal	Name	I/O	PU/PD(1)	Description
Power and Ground				
VDD (VCORE)	DSP Core Voltage	—	—	Dedicated supply pins powering the DSP core. Must be connected to +1.8 V.
VDDO (VIO)	I/O Voltage	—	—	Dedicated supply pins powering the I/O. Must be connected to +3.3 V.
VGNN (VESD)	5 V ESD Protection	—	—	Dedicated supply pins used to bias input protection diodes. If interfacing to other 5 V-powered devices, connect VGNN to +5 V; otherwise connect VGNN to +3.3 V.
GND	Ground	—	—	Common ground for ZipWireMulti device.
Clocks				
XTALI	Crystal Input	I	—	Crystal = 20.000 MHz (see Section 5.1)
XTALO	Crystal Output	O	—	Connection point for the crystal.
XTALO_B	Crystal Clock Out	O	—	Buffered-crystal oscillator output. 20.000 MHz.
EXT_CLK_REF_IN	Reference Clock Input	I	(4)	8 kHz reference clock input for DSP; this is used if the DSL clock needs to be locked to an 8 KHz network clock.
EXT_CLK_REF_OUT	Reference Clock Output	O	—	8 kHz reference clock output from DSP; this is recovered from the DSL clock and can be used as an 8 KHz network clock.
EXT_CLK_REF_SEL[0:2]	Reference Clock Select	I	PU(3)	Selects which of the 8 DSPs are used for EXT_CLK_REF_OUT. EXT_CLK_REF_SEL(2) is the most significant bit and EXT_CLK_REF_SEL(0) the least significant bit. The _DSL_EXT_CLK_REF_CONF API command must also be used to set the 8KHZ_CLK_SELECT and 8KHZ_OUT_EN parameters per the following table.
PLL_BYPASS#	—	I	PU(3)	Bypass internal PLL when active low.

EXT_CLK_REF_SEL PINS	8 KHZ_CLK_SELECT API	8 KHZ_OUT_EN API	EXT_CLK_REF_OUT SOURCE
000	0	ENABLE	DSP 0
001	1	ENABLE	DSP 1
010	2	ENABLE	DSP 2
011	3	ENABLE	DSP 3
100	4	ENABLE	DSP 4
101	5	ENABLE	DSP 5
100	6	ENABLE	DSP 6
101	7	ENABLE	DSP 7
111	X	DISABLE	DISABLED

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (2 of 7)

Signal	Name	I/O	PU/PD[1]	Description
PCM Interface				
RPCLK[0:7]	Receive PCM Clock	O	—	Clocks the PCM receive outputs: RPDAT and RPMSYNC. Normally derived by the internal clock recovery (DPLL). Can be derived by PEXTCLK or TPCLK. Rising edge or falling edge output transition are selectable.
PEXTCLK[0:7]	PCM External Clock	I	PU[3]	Optionally sources the RPCLK or TPCLK or both RPCLK and TPCLK.
RPDPLLCLK	RX PCM DPLL Clock	O	—	DPLL clock output from framer 0. This pin can be optionally used externally when the DPLL doesn't function as clock recovery but as clock generator.
RPDAT[0:7]	Receive PCM Data	O	—	During specified time slots, data is clocked out by RPCLK. This pin can be optionally three-stated during inactive time slots or when DSL Framer is bypassed.
RPMFSYNC[0:7]	Receive PCM MultiFrame Sync	O	—	Active high output from the receive time-base. Optionally, programmed to mark either Frame or Multi-Frame boundaries during framed application.
TPCLK[0:7]	Transmit PCM Clock	I	PU[3]	Normally, samples the PCM transmit inputs: TPDAT and TPMSYNC on the falling edge. The edge transition is selectable.
TPDAT[0:7]	Transmit PCM data	I	PU[3]	During specified time slots, data is sampled in by a selected clock source (TPCLK, PEXTCLK or DPLL Recovery clock).
TPMFSYNC[0:7]	Transmit PCM MultiFrame Sync	I/O	[5]	This signal resets the Transmit PCM Time Base during framed application and ignored in unframed mode. This signal internally delayed by a programmable bit and frame offset to coincide with TPDAT bit 0 frame 0. Optionally programmed to mark either Frame or Multi-Frame boundaries.
UTOPIA Interface				
ATM_RX_ADDR[0:4]	—	I	PU[3]	Functions as the address of the PHY device being selected for reception. Address 11111 (31 decimal) indicates a null PHY port.
ATM_RX_CLAV	—	O	—	Indicates a FIFO Empty condition or Cell Available condition. An external pull down resistor is required for this pin.
ATM_RX_DATA[0:15]	—	O	—	Outputs the received data to the ATM layer.
ATM_RX_ENB#	—	I	PU[3]	Enables data reception when asserted low.
ATM_RX_CLK	—	I	PU[3]	Clocks ATM_RX_DATA on the falling edge.
ATM_RX_PRTY	—	O	—	This pin is the parity calculated over the ATM_RX_DATA bus. Parity can be checked over ATM_RX_DATA [0:7] or ATM_RX_DATA [8:15]. This pin can represent either even or odd parity.
ATM_RX_SOC	—	O	—	Indicates the first byte of valid cell data transmitted when asserted high. This pin requires an external pull-down resistor.

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (3 of 7)

Signal	Name	I/O	PU/PD(1)	Description
ATM_TX_ADDR[0:4]	—	I	PU(3)	Functions as the address of the PHY device being selected for transmission. Address 11111 (31 decimal) indicates a null PHY port.
ATM_TX_CLAV	—	O	—	Indicates a FIFO Full condition or Cell Available condition. An external pull-down resistor is required for this pin.
ATM_TX_DATA[0:15]	—	I	PU(3)	Transmit data from the ATM layer.
ATM_TX_ENB#	—	I	PU(3)	Enables data transmission when asserted low.
ATM_TX_CLK	—	I	PU(3)	Used to sample ATM_TX_DATA on the falling edge.
ATM_TX_PRTY	—	I	PU(3)	This pin is the parity calculated over the ATM_TX_DATA bus. Parity can be checked over ATM_TX_DATA [0–7] or ATM_TX_DATA [8–15]. This pin can represent either even or odd parity.
ATM_TX_SOC	—	I	PU(3)	Indicates the first byte of valid cell data transmitted when asserted high.
ATM_ONE_SEC_IN	—	I	PU(3)	When asserted high, the device may latch and hold both status signals and counter values. This pin is typically strobed at one-second intervals. It is typically driven by ATM_ONE_SEC_OUT but can also be driven by an external one-second source.
ATM_ONE_SEC_OUT	—	O	—	When active high, this pin indicates that 8000 periods of the ATM_8_KHZ_IN input have passed. This pin is typically active at one-second intervals. This pin remains active for one period of the ATM_8_KHZ_IN pin. It typically drives ATM_ONE_SEC_IN.
ATM_8_KHZ_IN	—	I	PU(3)	This pin is a clock input used to derive ATM_ONE_SEC_OUT. This pin typically operates at a frequency of 8 kHz.
Framer Bypass Interface				
RXDAT[0:7]	DSP Receive Data	O	—	Data is clocked out on the rising edge of HXCLK.
HXP[0:7]	DSP Receive Symbol Alignment	O	—	Symbol clock that provides symbol boundary. Rising edge marks the least significant bit.
TXDAT[0:7]	DSP Transmit Data	I	PU(3)	Data is sampled on the falling edge of HXCLK. THDAT is aligned to the symbol boundary on HXP.
HXCLK[0:7]	DSP Data Rate Clock	O	—	This clock signal operates at the DSL data rate and controls the DSL interface data signals—HXP, RXDAT, RHAUX, RHMARK, TXDAT, THAUX, and THLOAD. Output data is clock out on the rising edge of HXCLK while input data is sampled on the falling edge of HXCLK.
Configuration Pins				
GPIO[0:7]	General Purpose I/O	I/O	PU(3)	General Purpose Input and Output. GPIO[0-6] = not defined GPIO[7]= Boot mode 0 = UIP (Serial) 1 = Host Bus

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (4 of 7)

Signal	Name	I/O	PU/PD(1)	Description
Reset				
RST#	Reset	I	PU(3)	Asynchronous active low input that places the device in an inactive state. This resets the DSP and internal ARM controller. This pin should be connected to the host processor's RST_OUT.
AFE Interface				
AFE_CLK[0:7]	—	O	—-	19–27 MHz AFE clock (always running). Connect directly to AFE AFE_CLK.
AFE_RST	—	O	—	Connect directly to AFE AFE_RST.
AFE_TX_SYNC[0:7]	—	O	—	Txdata strobe. Connect directly to AFE AFE_TX_SYNC.
AFE_RX_SYNC[0:7]	—	O	—	Rxdata strobe. Connect directly to AFE AFE_RX_SYNC.
AFE_TX_SER1[0:7]	—	O	—	Serial TX data sample one (LSB). Connect directly to AFE AFE_TX_SER1.
AFE_TX_SER2[0:7]	—	O	—	Serial TX data sample two (MSB). Connect directly to AFE AFE_TX_SER2.
AFE_RX_SER1[0:7]	—	I	(4)	Serial RX data sample one (LSB). Connect directly to AFE AFE_RX_SER1.
AFE_RX_SER2[0:7]	—	I	(4)	Serial RX data sample two (MSB). Connect directly to AFE AFE_RX_SER2.
AFE_CTRL_DATA[0:7]	—	I/O	(5)	Serial control data input/output. Connect directly to AFE AFE_CTRL_DATA.
Host Bus Interface				
HBA[2:8]	Host Bus Address	I	(4)	Host Bus Address bits 2–8. HBA(1) is not used because all the host bus registers are 16 bit registers aligned at the 32 bit boundary, and FIFO has 16 bit access capability. HBA(8) is used to decode register or FIFO access (Logic High to access FIFO and Logic Low to access registers).
HBD[0:15]	Host Bus Data	I/O	(5)	Host Bus Data bits 0–15. HBD(15) is the most significant bit and HBD(0) the least significant bit. In 8 bit bus mode, only HBD[7:0] signals are used, and HBD[15:8] signals should be pulled up.
HBBE[0:1]#	Host Bus Byte Enable	I	(4)	Host Bus Byte Enable. In 16 bit bus mode, HBBE1# = L indicates HBD[15:8] byte is valid on the current bus cycle, and HBBE0# = L indicates HBD[7:0] byte is valid on the current bus cycle. In 8 bit bus mode, HBBE1# should be connected to host bus HBA(0), and HBBE0# should be grounded.
HBADDWT#	Host Bus Add Wait State	I	PU(3)	Host Bus Add 1 wait state in Burst Mode. HBADDWT# = L adds one HBCLKIN to the host bus access in Burst mode, HPBURSTEN# = L, so each data is transferred in 2 HBCLKIN instead of every HBCLKIN clocks. In Non-burst mode, this signal does not have effect on the data transfer.

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (5 of 7)

Signal	Name	I/O	PU/PD(1)	Description
HPBURSTEN#	Host Bus Burst Enable	I	PU(3)	Host Bus Burst Enable. HPBURSTEN# = L to enable burst access to the CX28985; so multiple data can be transferred on single pulse of HPCS# = L. When HPBURSTEN# = H, the burst is not enabled and each HPCS# = L pulse has only one data transfer, either 8 bit or 16 bit.
HPCS#	Host Bus Chip Select	I	PU(3)	Host Bus Chip-Select. HPCS# = L selects this CX28985 to participate on the current bus cycle, and HPCS# = H tells the chip not to participate on the current bus cycle.
HBWRITE#	Host Bus Write	I	(4)	Host Bus Write. When HBWRITE# = H, the host bus cycle is a read cycle and the data is transferred from CX28995 to the host processor, and when HBWRITE# = L, the bus cycle is a write cycle and the data direction is opposite from the read cycle.
HBINT#	Host Bus External Interrupt	O	—	Active-low interrupt that signifies the API protocol is complete (see Section 14.4). HBINT# is active low and open drain so many chips can tie it together. An external pull up resistor is needed.
HBCLKIN	Host Bus Clock	I	(4)	Host Bus data, address and control signal clock. When HPINVCLK# = H the host bus signals are clocked at the rising edge of the HBCLKIN, when HPINVCLK# = L the host bus signals are clocked at the falling edge of the HBCLKIN.
HPINVCLK#	Host Bus Invert Clock	I	PU(3)	Host Bus invert clock. When HPINVCLK# = H the host bus signals are clocked at the rising edge of the HBCLKIN, when HPINVCLK# = L the host bus signals are clocked at the falling edge of the HBCLKIN. (Note: this is the strapping signal and should not be changed after system reset.)
Expansion Bus Interface				
EXP_ADR[0:19]	Expansion Bus Address	O	—	Expansion Bus Address bits 0–19. EXP_ADR(19) is the most significant bit and EXP_ADR(0) the least significant bit. EXP_A17 – EXP_A19 have an alternate function. The software can place these pins into a mode that can output signals (70 MHz) that external circuitry can use to generate an “eye pattern.” It is recommended to make these signals available via test points. EXP_A17 = clk_mux EXP_A18 = stp_a EXP_A19 = stp_sync_a_rx
EXP_DAT[0:31]	Expansion Bus Data	I/O	—	Expansion Bus Data bits 0–31. EXP_DAT(31) is the most significant bit and EXP_DAT(0) the least significant bit.
EXP_CLK	Expansion Bus Clock	O	—	Expansion Bus Clock In/Out.
EXP_CLK_EN	Expansion Bus Clock Enable	O	—	Expansion Bus Clock Enable. When logic low, this signal can be used to disable the clock to the SDRAM, to lower the power dissipation.
EXP_DQM[0:3]	Expansion Bus Mask/WE	O	—	Expansion Bus SDRAM Mask and SRAM Write Enable.

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (6 of 7)

Signal	Name	I/O	PU/PD[1]	Description
EXP_CAS#	Expansion Bus CAS	O	—	Expansion Bus SDRAM and SRAM Column Active Strobe.
EXP_RAS#	Expansion Bus RAS	O	—	Expansion Bus SDRAM and SRAM Row Active Strobe.
EXP_RD#	Expansion Bus Read	O	—	Expansion Bus read pulse.
EXP_WE#	Expansion Bus Write	O	—	Expansion Bus write.
EXP_CS[0:5]#	Expansion Bus CS	O	—	Expansion Bus Chip Select. EXP_CS0# = SDRAM/SRAM EXP_CS1# = SDRAM/SRAM EXP_CS2# = BOOT ROM EXP_CS3# = Peripheral 1 EXP_CS4# = Peripheral 2 EXP_CS5# = Peripheral 3
EXP_EXT_BOOT#	Expansion Bus Ext Boot ROM	I	PU[3]	Use external Boot ROM on the Expansion Bus when active low.
Serial Port Interface				
UART_RX	RS232 Rx Data	I	PU[3]	UIP interface Receive Data.
UART_TX	RS232 Tx Data	O	—	UIP interface Transmit Data.
JTAG Interface				
TRST#	Test Port reset	I	PU[3]	Active-low resets the TAP controller. This pin should be connected high for normal operation.
TDI	Test data In	I	PU[3]	JTAG test data input per *IEEE Std. 1149.1-1990*. Used for loading all serial instructions and data into internal test logic. Sampled on the rising edge of TCK. TDI can be left unconnected when not being used because it is internally pulled high.
TDO	Test Data Out	O	—	JTAG test data input per *IEEE Std. 1149.1-1990*. Three-state output used for reading all serial configuration and test data from internal test logic. Updated on the falling edge of TCK.
TCK	Test Clock	I	PU[3]	JTAG test data input per *IEEE Std. 1149.1-1990*. Used for all test interface and internal test logic operations. If unused, TCK should be pulled low.
TMS	Test Mode Select	I	PU[3]	JTAG test data input per IEEE Std. 1149.1-1990. Input signal used to control the test-logic state machine. Sampled on the rising edge of TCK. TMS can be left unconnected when not being used because it is internally pulled high.
Miscellaneous and Test Modes				
TEST_SM	—	I	PD[2]	—
TEST_SE[0:2]	—	I	PD[2]	—
TEST_ST#	—	I	PU[3]	—

Table 6-6. ZipWireMulti Transceiver/Framer Signal Definitions (7 of 7)

Signal	Name	I/O	PU/PD(1)	Description
TEST_TM[0:2]	—	I	PU(3)	—
TEST_CLK	—	I	PU(3)	—
3_STATE#	—	I	PU(3)	All outputs are high impedance when active low.

NOTE(S):
(1) Internally Pulled-Up (PU) or Pulled-Down (PD) with a 15–75 KΩ resistor.
(2) When pin is not used, it should left unconnected or externally tied low.
(3) When pin is not used, it should left unconnected or externally tied high.
(4) When pin is not used, it should externally tied low or high.
(5) When pin is not used, it should externally tied low or high through a resister.

6.2.3 ZipWireMulti AFE Signal Descriptions

Table 6-7 lists the ZipWireMulti AFE signal (pin) descriptions.

Table 6-7. ZipWireMulti AFE Signal Descriptions (1 of 2)

Signal Name	I/O	Description
Power and Ground		
VA12	—	+12 V Analog Supply
A12GND	—	Analog Ground for +12 V Analog Supply
VA33	—	+3.3 V Analog Supply
A33GND	—	Analog Ground for +3.3 V Analog Supply
VD33	—	+3.3 V Digital I/O Supply
DGND	—	Digital I/O Ground
Transmit Section		
LDOP	O	Transmit, Positive (+) Line Driver Output
LDON	O	Transmit, Negative (–) Line Driver Output
LDAOP	I	Transmit, Auxiliary (+) Line Driver Input
LDAON	I	Transmit, Auxiliary (–) Line Driver Input
RCDRVP	O	Transmit, Positive (+) RC Driver Output
RCDRVN	O	Transmit, Negative (–) RC Driver Output
Transmit References		
VRPTX	REF	Transmit, Positive (+) Voltage Reference
VRNTX	REF	Transmit, Negative (–) Voltage Reference
Transmit Internal Signal		
LD_ML	—	Internal Line Driver Signal
LD_MH	—	Internal Line Driver Signal
LD_PH	—	Internal Line Driver Signal
LD_PL	—	Internal Line Driver Signal
Receive Section		
VXP	I	Receive, Positive (+) Transformer Line Input
VXN	I	Receive, Negative (–) Transformer Line Input
VHP	I	Receive, Positive (+) Hybrid Analog Input
VHN	I	Receive, Negative (–) Hybrid Analog Input
Receive References		
VRPRX	REF	Receive, Positive (+) Voltage Reference
VRNRX	REF	Receive, Negative (–) Voltage Reference
VBGP	REF	Reference, Positive (+) Band-gap Reference
VBGN	REF	Reference, Negative (–) Band-gap Reference

Table 6-7. ZipWireMulti AFE Signal Descriptions (2 of 2)

Signal Name	I/O	Description
VCMO	REF	Reference, Output Common Mode Voltage
VCMI	REF	Reference, Input Common Mode Voltage
AVBIAS	REF	Reference, Compensation Capacitor
RBIAS	REF	Reference, Current Reference Resistor
DSP Interface		
AFE_CLK	I	19–27 MHz AFE clock (always running). Connect directly to DSP AFE_CLK.
AFE_RX_SYNC	I	Rxdata strobe. Connect directly to DSP AFE_RX_SYNC.
AFE_TX_SYNC	I	Txdata strobe. Connect directly to DSP AFE_TX_SYNC.
AFE_RST	I	Connect directly to DSP AFE_RST.
AFE_CTRL_DATA	I/O	Serial control data input/output. Connect directly to DSP AFE_CTRL_DATA.
AFE_RX_SER1	O	Serial RX data sample one (LSB). Connect directly to DSP AFE_RX_SER1.
AFE_RX_SER2	O	Serial RX data sample two (MSB). Connect directly to DSP AFE_RX_SER2.
AFE_TX_SER1	I	Serial TX data sample one (LSB). Connect directly to DSP AFE_TX_SER1.
AFE_TX_SER2	I	Serial TX data sample two (MSB). Connect directly to DSP AFE_TX_SER2.
Miscellaneous and Test		
SCAN_EN	I	ASIC scan enable input. Connect to logic low (GND).

7.0 Electrical and Mechanical Specifications

The following specifications apply to both the ZipWireMulti Transceiver/Framer and the ZipWireMulti AFE.

7.1 Specifications for the ZipWireMulti Transceiver/Framer and ZipWireMulti AFE

7.1.1 Recommended Operating Conditions

Table 7-1. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
VFE 12.0 V Analog Supply	VA12	11.4	12.0	12.60	V
DSP 3.3 V Input/Output Supply	VIO	3.15	3.30	3.45	V
AFE 3.3 V Analog Supply	VA33	3.15	3.30	3.45	V
AFE 3.3 V Digital Supply	VD33	3.15	3.30	3.45	V
DSP 1.8 V Digital Core	Vdd	1.60	1.80	2.00	V
NOTE(S): A12GND = A33GND = GND = DGND = 0 V; other voltages with respect to 0 V.					

7.1.2 Absolute Maximum Ratings

Table 7-2. Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
VA12	VA12	—	12.0	—	V
VIO	VIO	–0.3	3.3	4.6	V
VA33	VA33	–0.3	3.30	3.45	—
VD33	VD33	–0.3	3.30	3.45	
VDD	VDD	–0.3	1.8	2.5 V	V
Voltage on any Signal Pin	—	GND –0.3	—	VIO + 0.3	V
Input Current, any pin except supplies	IMAX	—	—	±10	mA
Analog Input Voltage	—	–0.3	—	VAA + 0.3	V
Digital Input Voltage for Transceiver and Framer	—	–0.3	—	V_{GNN} + 0.3	V
Digital Input Voltage for AFE	—	–0.3	—	3.45	V
Ambient Operating Temperature	T_A	–40	25	+85	°C
Junction Temperature	T_J	—	—	125	°C
Storage Temperature (ambient)	T_{SA}	–65	—	150	°C
Soldering Temperature	TSOL	–65	—	260	°C
Vapor Phase Soldering	TVSOL	–65	—	220	°C
Air Flow	0	—	—	—	L.F.P.M.

NOTE(S): Operation beyond these limits may cause permanent damage to the device. Normal operation is not guaranteed at these extreme conditions.

7.2 Thermal Characteristics

7.2.1 ZipWireMulti AFE

For the 7 × 7 mm LGA (AFE) with 0 m/s of airflow, θ_{JA}~ 43 °C/W.

7.2.2 ZipWireMulti Transceiver/Framer

For the 27 × 27 mm PBGA (Transceiver/Framer) with 0 m/s of airflow, θ_{JA}~ 18 °C/W.

7.3 Specifications for ZipWireMulti Transceiver/Framer Only

7.3.1 Power Dissipation

The 8-port ZipWireMulti Transceiver/Framer 1.8 V power dissipation is 1.8 watts.
The 8-port ZipWireMulti Transceiver/Framer 3.3 V power dissipation is 1.1 watts.

7.3.2 DC Characteristics

Table 7-3 lists transceiver framer DC characteristics.

Table 7-3. Transceiver/Framer DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Digital Inputs					
Input High Voltage	VIH	2.0	—	V_{GNN} +0.25	V
Input Low Voltage	VIL	0	—	0.80	V
Input Leakage Current	IIL/IIH	—	—	—	µA
Input Capacitance	CIN	—	—	—	pF
Digital Outputs					
Output High Voltage	VOH	2.4	—	—	V
Output Low Voltage	VOL	—	—	0.40	V
Three-State Output Leakage	ILK	—	—	—	µA
Output Capacitance	C_{OUT}	—	—	—	pF
Digital Bidirectionals					
Three-State Output Leakage	ILK	—	—	—	µA
Input/Output Capacitance	CINOUT	—	—	—	pF

7.3.3 Host Bus AC Timing

On host bus read cycles, the first data access of a burst read or any data access of a non-burst read, HBD will be valid when Tadr, Tdoen, and Tdv are met. On the second or later data access of a burst ready cycle from the FIFO, HBA[8] = 1 and HBBE# no changed, HBD will be valid after Tckdr is met. On a host bus write cycle, only Tsetup and Thold must be observed. Also, HBADDWT# must meet Tsetup and Thold for both host bus read and write.

Figures 7-1 through 7-3 illustrate AC Timing Diagrams.

Figure 7-1. Host Processor Read Timing

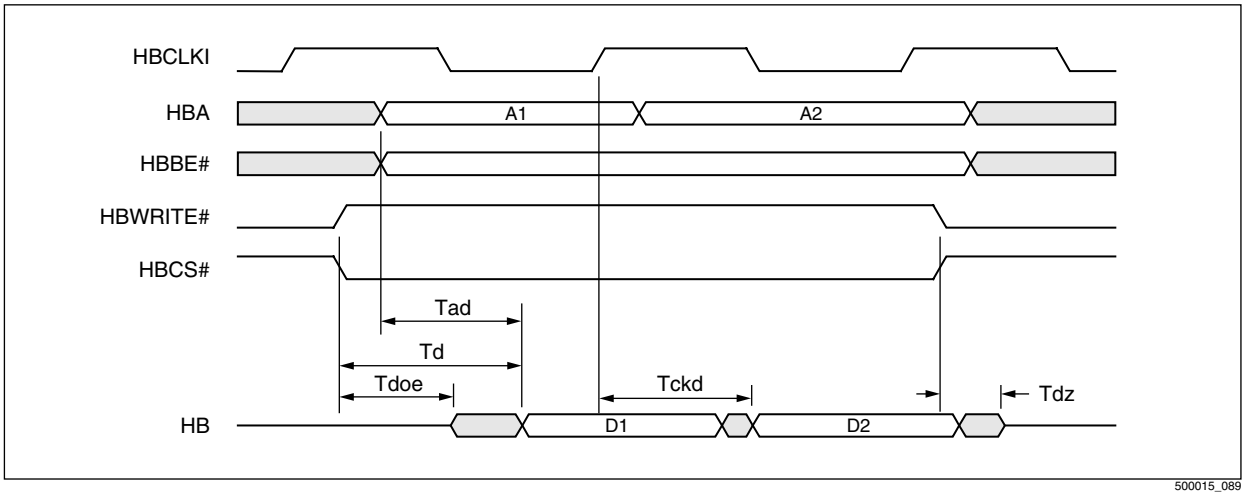


Figure 7-2. Host Bus Signals Timing Relative to HBCLKIN (HBINVCLK# = H) (For Read and Write Cycles)

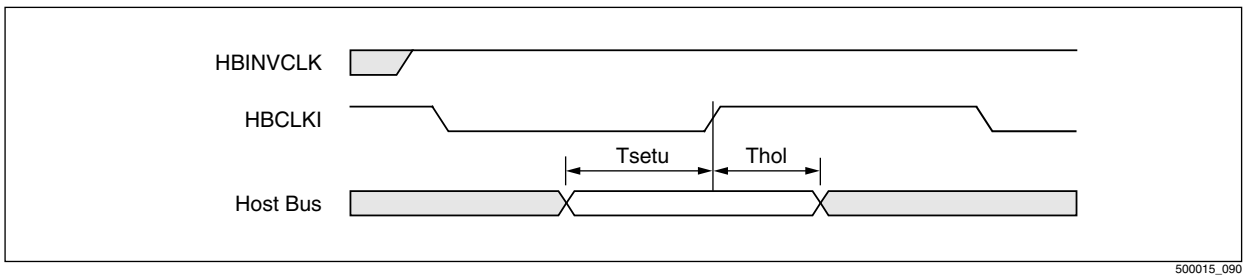


Figure 7-3. Host Bus Signals Timing Relative to HBCLKIN (HBINVCLK# = L) (For Read and Write Cycles)

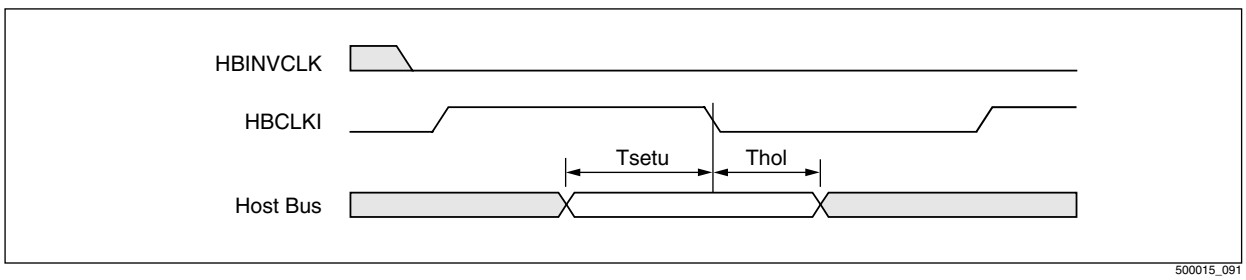


Table 7-4 lists AC Timing Parameters.

Table 7-4. Host Processor Read Timing

Parameter	Description	Minimum	Maximum with 25 pf/50 pf load
Tadr	HBA and HBBE# valid to HBD valid on the first data access of a host bus read. (Assume Tdv meets the timing)	—	18 ns/20 ns
Tckdr	HBCLKIN active edge to HBD valid on the second and later data access of the host bus burst read from the FIFO.	—	19 ns/23 ns
Tdoen	HBWRITE# = H and HBCS# = L to HBD start driving the bus.	9 ns	—
Tdv	HBWRITE# = H and HBCS# = L to HBD valid on the first data access of a host bus read. (Assume Tadr meets the timing).	—	13 ns/15 ns
Tdz	HBWRITE# = L or HBCS# = H to HBD three-stated.	—	13 ns/16 ns
Tsetup	Host bus signals to HBCLKIN active edge set up time.	5 ns	—
Thold	Host bus signals to HBCLKIN active edge hold time.	0 ns	—

NOTE(S):
1. The active edge is the rising edge when HBINVCLK# = H, and the falling edge when HBINVCLK# = L.
2. Host bus signals referred at Tsetup and Thold descriptions are: HBA, HBBE#, HBD, HBCS#, HBWRITE#, and HBADDWT#.
3. On a host bus write cycle, only Tsetup and Thold must be observed.

7.3.4 DSL Framer Timing Requirements

Figure 7-4. Input Clock Requirements

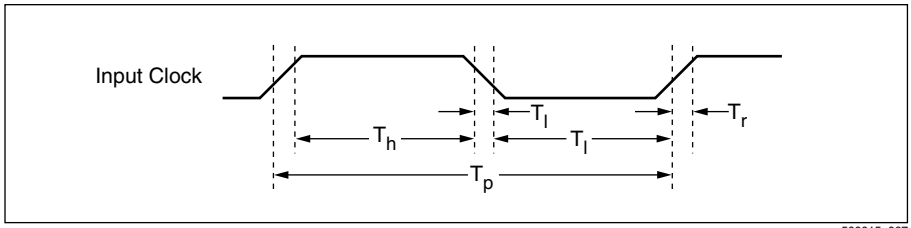


Table 7-5. Input Clock Requirements

Symbol	Parameter	Minimum	Maximum	Units
—	TPCLK, PEXTCLK	—	—	—
1/Tp	Frequency	0.064	18.432	MHz
Th	Clock Width High	0.4 × Tp	0.6 × Tp	ns
T1	Clock Width Low	0.4 × Tp	0.6 × Tp	ns
Tr	Clock Rise Time	—	20	ns
Tf	Clock Fall Time	—	20	ns

Figure 7-5. Input Setup and Hold Requirements

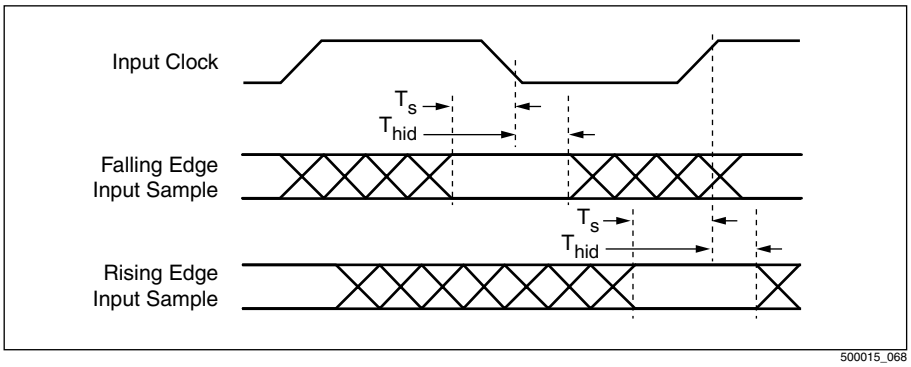


Table 7-6. Input Setup and Hold Requirements

Symbol	Parameter	Minimum	Maximum	Units
—	TPDAT, TPINSDAT, TPMFSYNC	—	—	—
Ts	Input Setup Time	35	—	ns
Thld	Input Hold Time	10	—	ns

7.3.5 DSL Framer Switching Characteristics

Figure 7-6. Output Characteristics

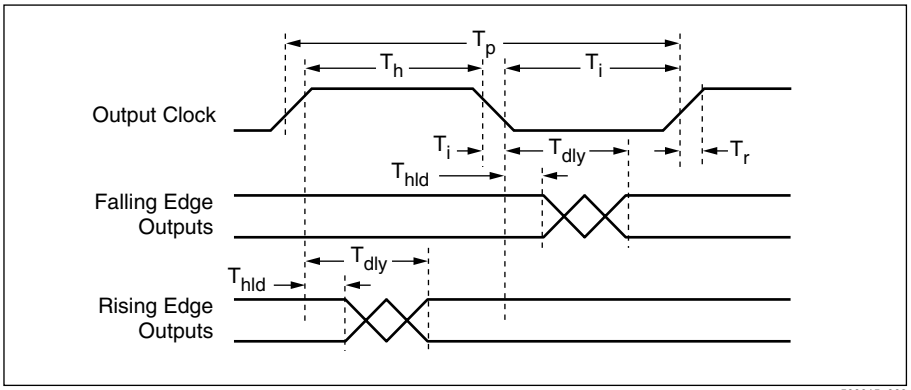


Table 7-7. Output Characteristics

Symbol	Parameter	Minimum	Maximum	Units
—	RPCLK, RPDPLLCLK	—	—	—
1/Tp	Frequency	0.064	18.432	MHz
Th	Clock Width High	Tp / 2 – 20	Tp / 2 + 20	ns
T1	Clock Width Low	Tp / 2 – 20	Tp / 2 + 20	ns
Tr	Clock Rise Time	—	15	ns
Tf	Clock Fall Time	—	15	ns
—	RPDAT, RPDROP, RPMFSYNC, TPMFSYNC	—	—	—
Thld	Output Data Hold	0	—	ns
Tdly	Output Data Delay	—	25	ns

7.3.6 UTOPIA Interface Timing

Figures 7-7 through 7-8 and Tables 7-8 through 7-9 show the timing requirements and characteristics of the UTOPIA interface.

Figure 7-7. UTOPIA Transmit Timing Diagram

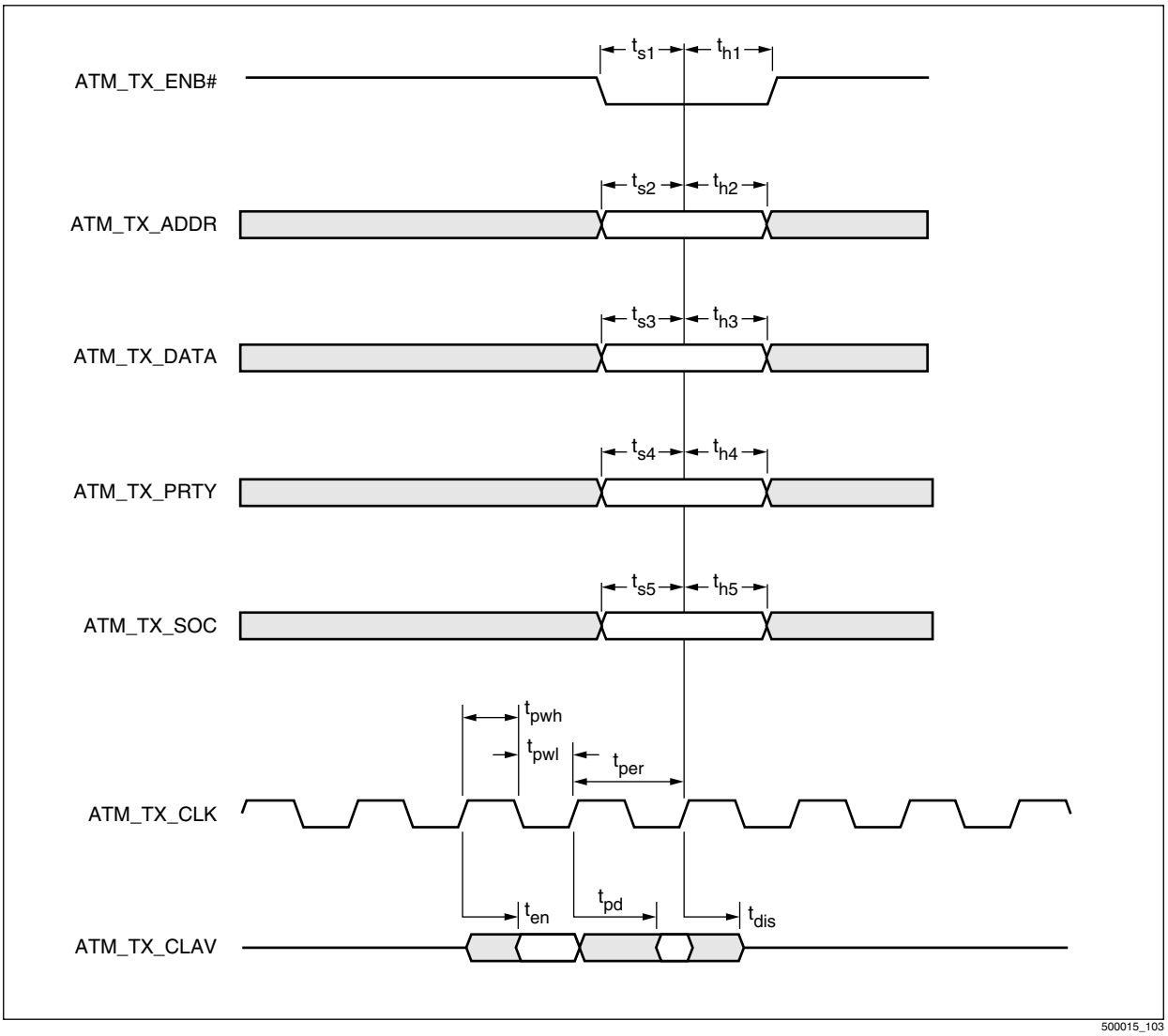


Table 7-8. UTOPIA Transmit Timing Table

Label	Description	Min	Max	Unit
t_{pwl}	Pulse Width Low, ATM_TX_CLK	8	—	ns
t_{pwh}	Pulse Width High, ATM_TX_CLK	8	—	ns
t_{per}	Period, ATM_TX_CLK	20	—	ns
t_{s1}	Setup, ATM_TX_ENB# to the rising edge of ATM_TX_CLK	4	—	ns
t_{h1}	Hold, ATM_TX_ENB# from the rising edge of ATM_TX_CLK	1	—	ns
t_{s2}	Setup, ATM_TX_ADDR to the rising edge of ATM_TX_CLK	4	—	ns
t_{h2}	Hold, ATM_TX_ADDR from the rising edge of ATM_TX_CLK	1	—	ns
t_{s3}	Setup, ATM_TX_DATA to the rising edge of ATM_TX_CLK	4	—	ns
t_{h3}	Hold, ATM_TX_DATA from the rising edge of ATM_TX_CLK	1	—	ns
t_{s4}	Setup, ATM_TX_PRTY to the rising edge of ATM_TX_CLK	4	—	ns
t_{h4}	Hold, ATM_TX_PRTY from the rising edge of ATM_TX_CLK	1	—	ns
t_{s5}	Setup, ATM_TX_SOC to the rising edge of ATM_TX_CLK	4	—	ns
t_{h5}	Hold, ATM_TX_SOC from the rising edge of ATM_TX_CLK	1	—	ns
t_{en}	Enable, ATM_TX_CLAV from the rising edge of ATM_TX_CLK	1	4	ns
t_{pd}	Propagation Delay, ATM_TX_CLAV from the rising edge of ATM_TX_CLK	1	9	ns
t_{dis}	Disable, ATM_TX_CLAV from the rising edge of ATM_TX_CLK	1	4	ns

Figure 7-8. UTOPIA Receive Timing Diagram

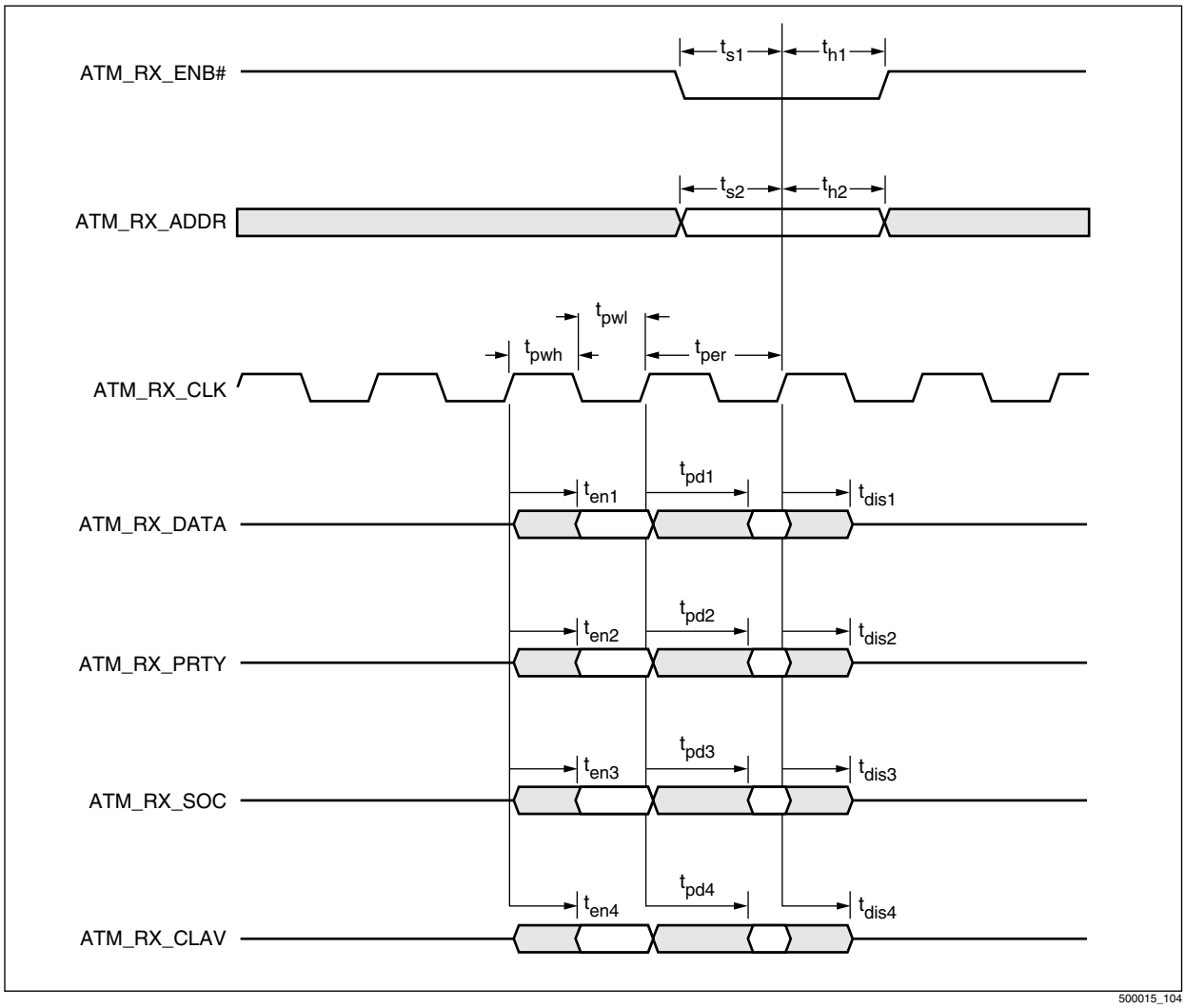


Table 7-9. UTOPIA Receive Timing Table

Label	Description	Min	Max	Unit
t_{pwl}	Pulse Width Low, ATM_RX_CLK	8	—	ns
t_{pwh}	Pulse Width High, ATM_RX_CLK	8	—	ns
t_{per}	Period, ATM_RX_CLK	20	—	ns
t_{s1}	Setup, ATM_RX_ENB# to the rising edge of ATM_RX_CLK	4	—	ns
t_{h1}	Hold, ATM_RX_ENB# from the rising edge of ATM_RX_CLK	1	—	ns
t_{s2}	Setup, ATM_RX_ADDR to the rising edge of ATM_RX_CLK	4	—	ns
t_{h2}	Hold, ATM_RX_ADDR from the rising edge of ATM_RX_CLK	1	—	ns
t_{en1}	Enable, ATM_RX_DATA[15:0] from the rising edge of ATM_RX_CLK	2	10	ns
t_{pd1}	Propagation Delay, ATM_RX_DATA[15:0] from the rising edge of ATM_RX_CLK	1	14	ns
t_{dis1}	Disable, ATM_RX_DATA[15:0] from the rising edge of ATM_RX_CLK	2	10	ns
t_{en2}	Enable, ATM_RX_PRTY from the rising edge of ATM_RX_CLK	2	10	ns
t_{pd2}	Propagation Delay, ATM_RX_PRTY from the rising edge of ATM_RX_CLK	1	14	ns
t_{dis2}	Disable, ATM_RX_PRTY from the rising edge of ATM_RX_CLK	2	10	ns
t_{en3}	Enable, ATM_RX_SOC from the rising edge of ATM_RX_CLK	2	10	ns
t_{pd3}	Propagation Delay, ATM_RX_SOC from the rising edge of ATM_RX_CLK	1	14	ns
t_{dis3}	Disable, ATM_RX_SOC from the rising edge of ATM_RX_CLK	2	10	ns
t_{en4}	Enable, ATM_RX_CLAV from the rising edge of ATM_RX_CLK	1	8	ns
t_{pd4}	Propagation Delay, ATM_RX_CLAV from the rising edge of ATM_RX_CLK	1	8	ns
t_{dis4}	Disable, ATM_RX_CLAV from the rising edge of ATM_RX_CLK	1	8	ns

7.4 AC Electrical Specifications

7.4.1 Expansion Bus Timing

7.4.2 SDRAM Timing Specification

The timing of the SDRAM signals depends on the load applied to the output signals and operating conditions. The following timing data is guaranteed under worst case operating conditions and 15 pf load on signals EXP_CLK, EXP_RAS#, EXP_CAS#, EXP_DQMs and EXP_CS, and 50 pf on Address, Data and EXP_WE#, because these are shared buses.

Figure 7-9. SDRAM Interface Timing Diagram

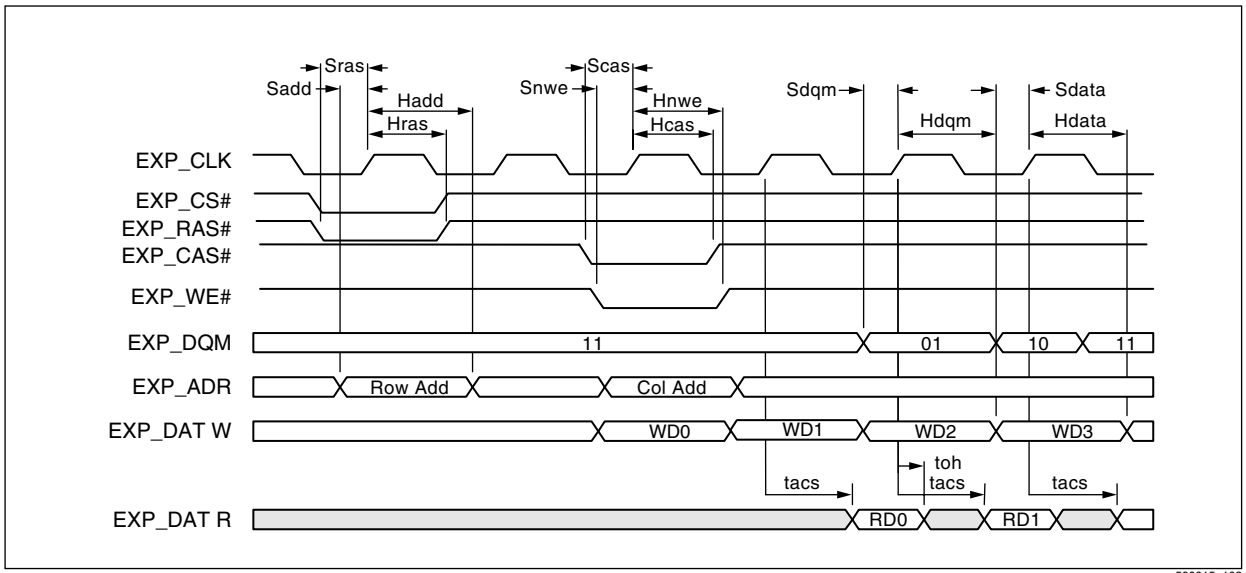


Table 7-10. SDRAM Interface Timing Parameters

Parameter	Comment	Min	Max	Reference
Sras	EXP_RAS# Setup time	3 ns	—	EXPCLK rising
Hras	EXP_RAS# Hold time	2 ns	—	EXPCLK rising
Scas	EXP_CAS# Setup time	3 ns	—	EXPCLK rising
Hcas	EXP_CAS# Hold time	2 ns	—	EXPCLK rising
Scs	EXP_CS# Setup time	3 ns	—	EXPCLK rising
Hcs	EXP_CS# Hold time	2 ns	—	EXPCLK rising
Snwe	EXP_WE# Setup time	3 ns	—	EXPCLK rising
Hnwe	EXP_WE# Hold time	2 ns	—	EXPCLK rising
Sdqm	EXP_DQM Setup time	3 ns	—	EXPCLK rising
Hdqm	EXP_DQM Hold time	2 ns	—	EXPCLK rising
Sdata	EXP_DAT Setup time	3 ns	—	EXPCLK rising
Hdata	EXP_DAT Hold time	2 ns	—	EXPCLK rising
Sadd	EXP_ADR Setup time	3 ns	—	EXPCLK rising
Hadd	EXP_ADR Hold time	2 ns	—	EXPCLK rising
tacs	SDRAM DQ Access time	—	8 ns	EXPCLK rising
toh	SDRAM DQ Hold time	2 ns	—	EXPCLK rising

7.5 Specifications for ZipWireMulti AFE Only

The following specifications apply only to the ZipWireMulti AFE.

7.5.1 Power Consumption

Table 7-11 shows the breakdown for the Single Channel ZipWireMulti AFE power consumption for 1,552 kbps G.shdsl Enhanced Performance Asymmetric PSD (EPAP) and 2,320 kbps G.shdsl Symmetric PSD mode. Power consumption for other data rates will be similar.

These power values assume normal operating modes of random (scrambled) data with a 10 Kft. 26 AWG line

The Max values are for worst case temperature, voltage, and silicon process.

The power consumption includes the power delivered to the DSL line. In EPAP mode, the power delivered to the line is ~95 mW. In Symmetric PSD mode, the power delivered to the line is ~56 mW.

Table 7-11. ZipWireMulti AFE Power Dissipation

Parameter	Condition	Symbol	Min	Typ	Max	Units
AFE, +12.0 V	EPAP	$PD_{AFE\ 12.0}$	—	450	—	mW
	Sym PSD		—	TBD	—	mW
AFE, +3.3 V	—	$PD_{AFE\ 3.30}$	—	420	—	mW

7.5.2 DC Characteristics

Table 7-12. AFE DC Characteristics (1 of 2)

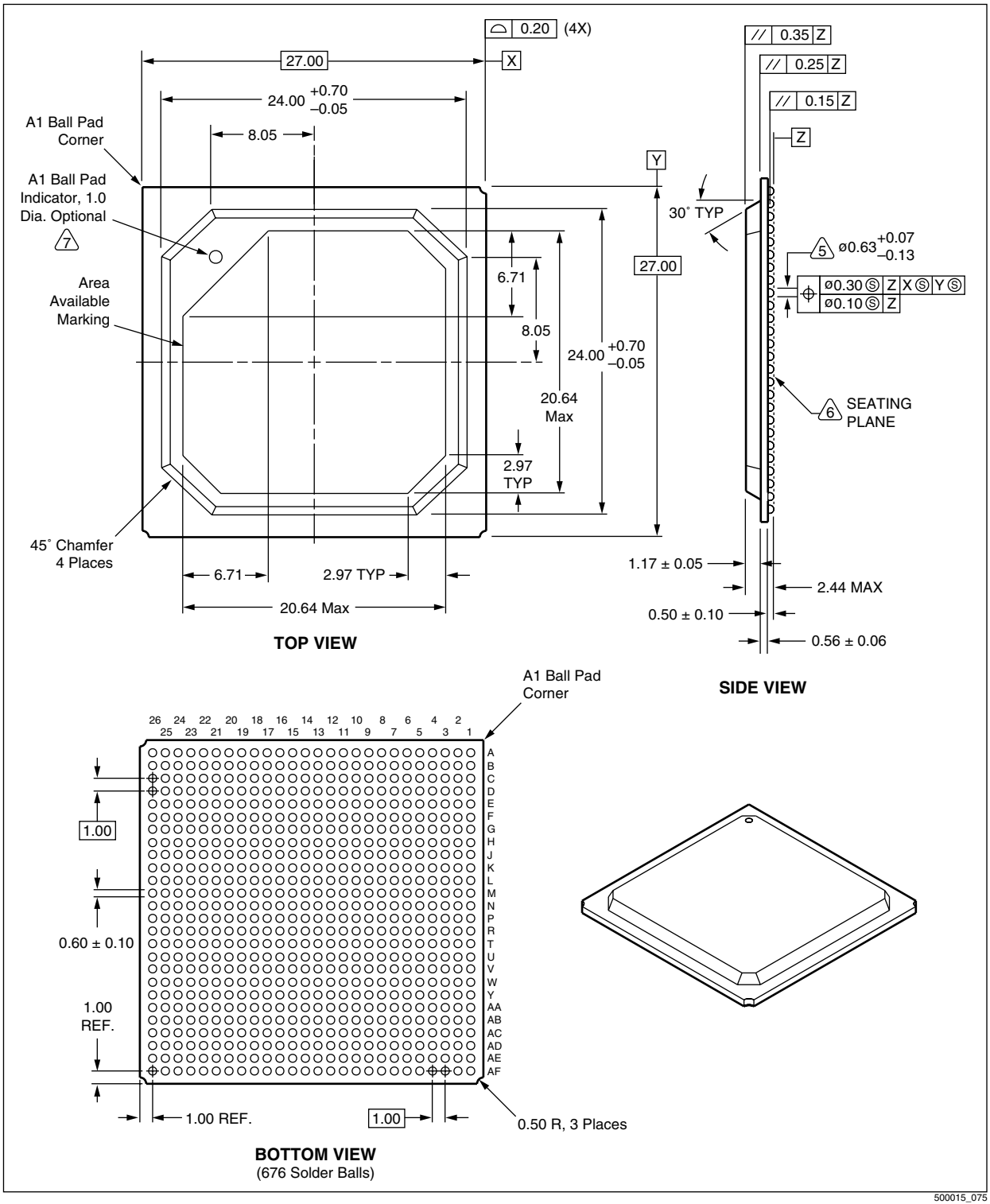
Parameter	Symbol	Min	Typ	Max	Units
Digital Inputs					
Input High Voltage	VIH	2.50	—	3.45	V
Input Low Voltage	VIL	GND	—	0.4 V	V
Input Leakage Current	IIL/IIH	–10	—	10	µA
Input Capacitance	CIN	—	2.9	—	pF
Digital Outputs					
Output High Voltage	VOH	0.9 VIO	—	VIO	V

Table 7-12. AFE DC Characteristics (2 of 2)

Parameter	Symbol	Min	Typ	Max	Units
Output Low Voltage	VOL	GND	—	0.1 VIO	V
Three-State Output Leakage	ILK	10	—	10	µA
Output Capacitance	CIN	—	3.1	—	pF
Digital Bidirectionals					
Three-State Output Leakage	ILK	–10	—	10	µA
Input/Output Capacitance	CINOUT	—	3.0	—	pF

7.6 Mechanical Specifications

Figure 7-10. Package Outline for 27 × 27 mm, 676-Pin Plastic Ball Grid Array (PBGA)



7.6.1 The 7 × 7 mm LGA

7.6.1.1 Purpose

The LGA package features a laminate substrate with LGA perimeter pads and an exposed ground pad. The center pad improves the thermal and electrical performance of the package. To make optimum use of these performance improvements, the PWB must be designed with this technology in mind. This application note will focus on the specifics of integrating the LGA into the PWB design.

7.6.1.2 Ground Pad Geometry

To take advantage of the LGA performance improvements, a solder-tinned-copper pad with thermal vias is required on the PWB. The pad size should be equal to the LGA body size minus 2.0 mm with a solder mask opening equal to the ground pad of the LGA. The ground pad dimension can be found on the device technical specification sheet. The result is a solder mask-defined center pad.

An array of 0.33 mm diameter thermal vias plated with 1 oz. copper should be placed on the ground pad and shorted to the ground plane of the PWB. This thermal via pattern represents a copper cross section in the barrel of the thermal via of approximately 1% of the total ground pad area. For the exposed region of the ground pad, if the plating is not thick enough to effectively plug the barrel of the via when plated, then use a solder mask to cap the vias with a dimension equal to the via diameter + 0.1 mm minimum. This prevents solder from being wicked through the thermal via and potentially creating a solder void in the region between the package bottom and the ground pad on the surface of the PWB.

Figure 7-11 illustrates an example of both metallization (lower left) and solder mask (lower right) for the 5 × 5 LGA.

Figure 7-11. Recommended Feature Sizes for LGA Footprint

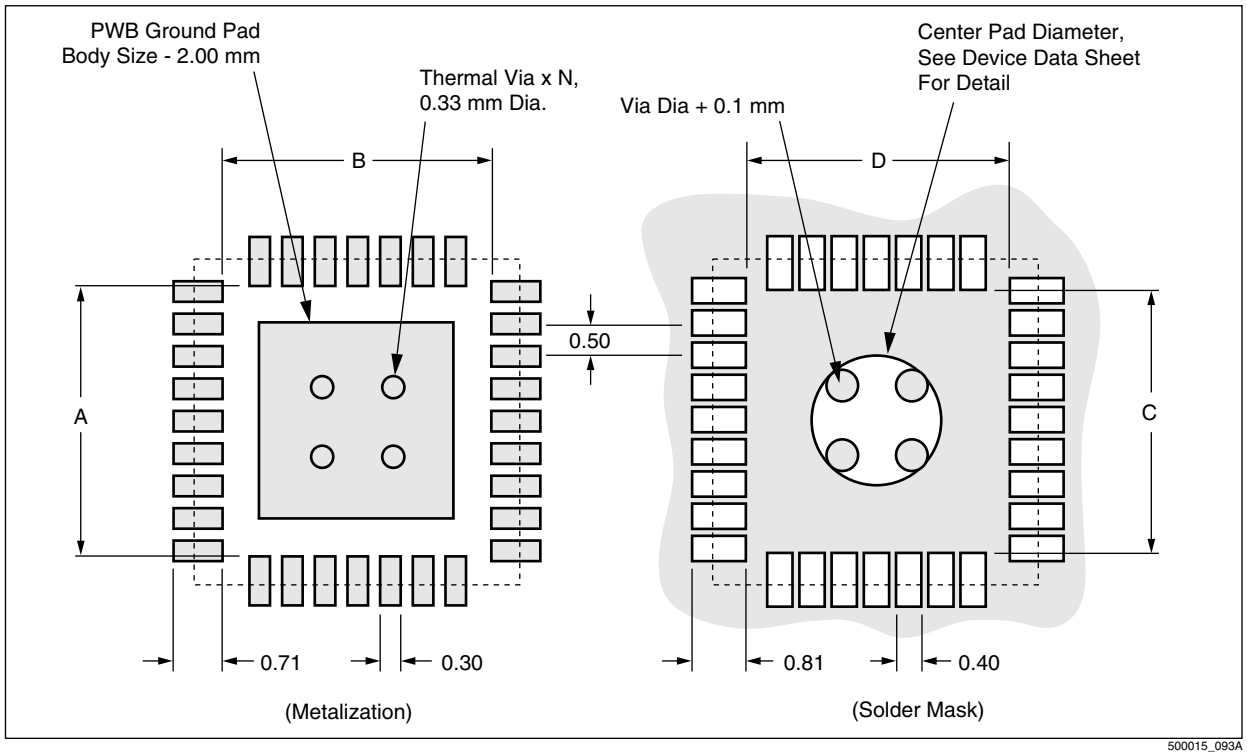


Table 7-13 lists the dimensions and thermal via array for the square and rectangular LGA package family.

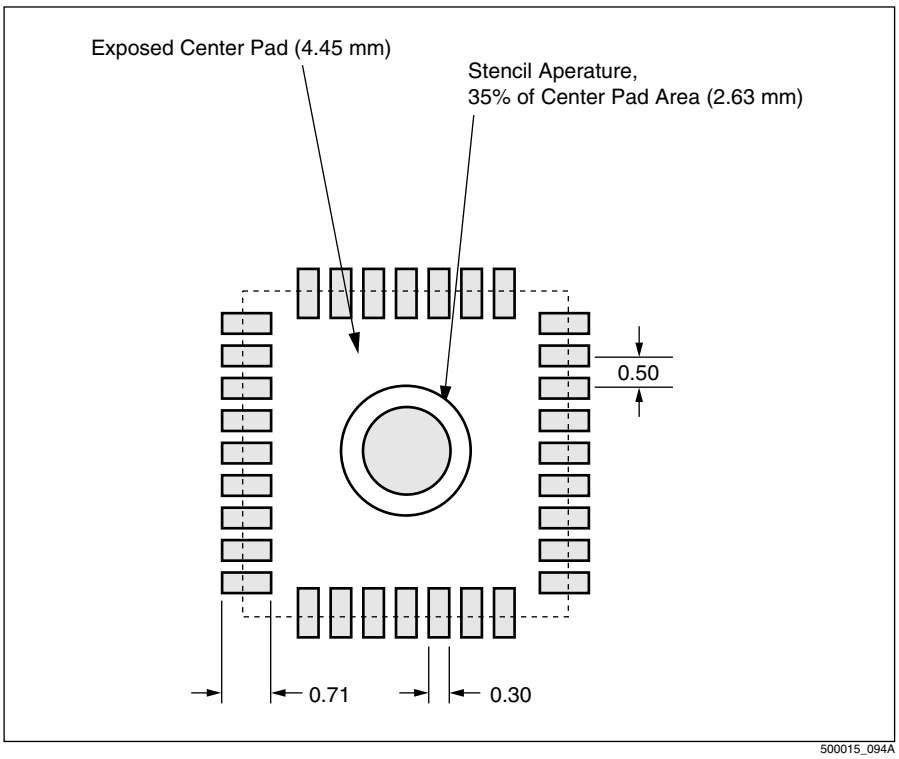
Table 7-13. Dimensional Parameters, Square Packages (mm)

Package Type	X/Y Pin Count	A	B	C	D	N(1)
48 pin 7 × 7	11/13	6.20	6.20	6.10	6.10	3 × 7, 9
NOTE(S): (1)N represents the total number of thermal vias to be placed evenly across the entire PWB center pad.						

7.6.1.3 Solder Stencil Determination

Solder and solder paste volume control is critical for SMT assembling of LGA packages onto the PWB. Stencil thickness and aperture openings should be optimized according to the optimal solder volume. In general, LGA packages can be reflowed on boards using a range of stencil thickness from 4–6 mils. Stencil thickness < 4 mils should be avoided to prevent insufficient solder joint volume. To obtain a reasonable standoff, it is suggested to make the aperture opening for the LGA peripheral pads identical to the size of the peripheral metallization shown in Figure 7-12 By extending the pads beyond the edge of the package, a solder joint fillet is achieved which can be visually inspected. In addition to this, the extended pad configuration also improves the solder paste printability.

Figure 7-12. Recommended Feature Sizes for Stencil Aperture



In general, as the stencil thickness increases, the aperture opening for the center pad must be reduced. To obtain a preferred center pad solder joint without solder balling and to maintain contact on the LGA pads, reduce the total amount of solder volume as indicated in Table 7-14. The stencil aperture diameter represents 35% of the actual center pad area. For center pad diameters not listed in Table 7-14 the diameter of the stencil aperture can be calculated by taking 35% of the actual center pad area.

Table 7-14. Recommended Stencil Aperture for the Center Pad (mm)

A (Center Pad Diameter)	B (Stencil Aperture Diameter)
4.45	2.63

7.6.1.4 Moisture Sensitivity

Conexant device manufacturing procedures minimize the effects of moisture absorption and associated problems. Device packages that are sensitive to moisture absorption are baked and vacuum packed for shipping with a desiccant and a humidity indicator card. It is recommended that procedures for handling moisture sensitive devices follow IPC (Institute for Printed Circuits) standards. The general rule is to bake the parts prior to reflow for 24 hours at 125 °C if the dry bag has been opened and the parts have been exposed to open air for 72 hours.

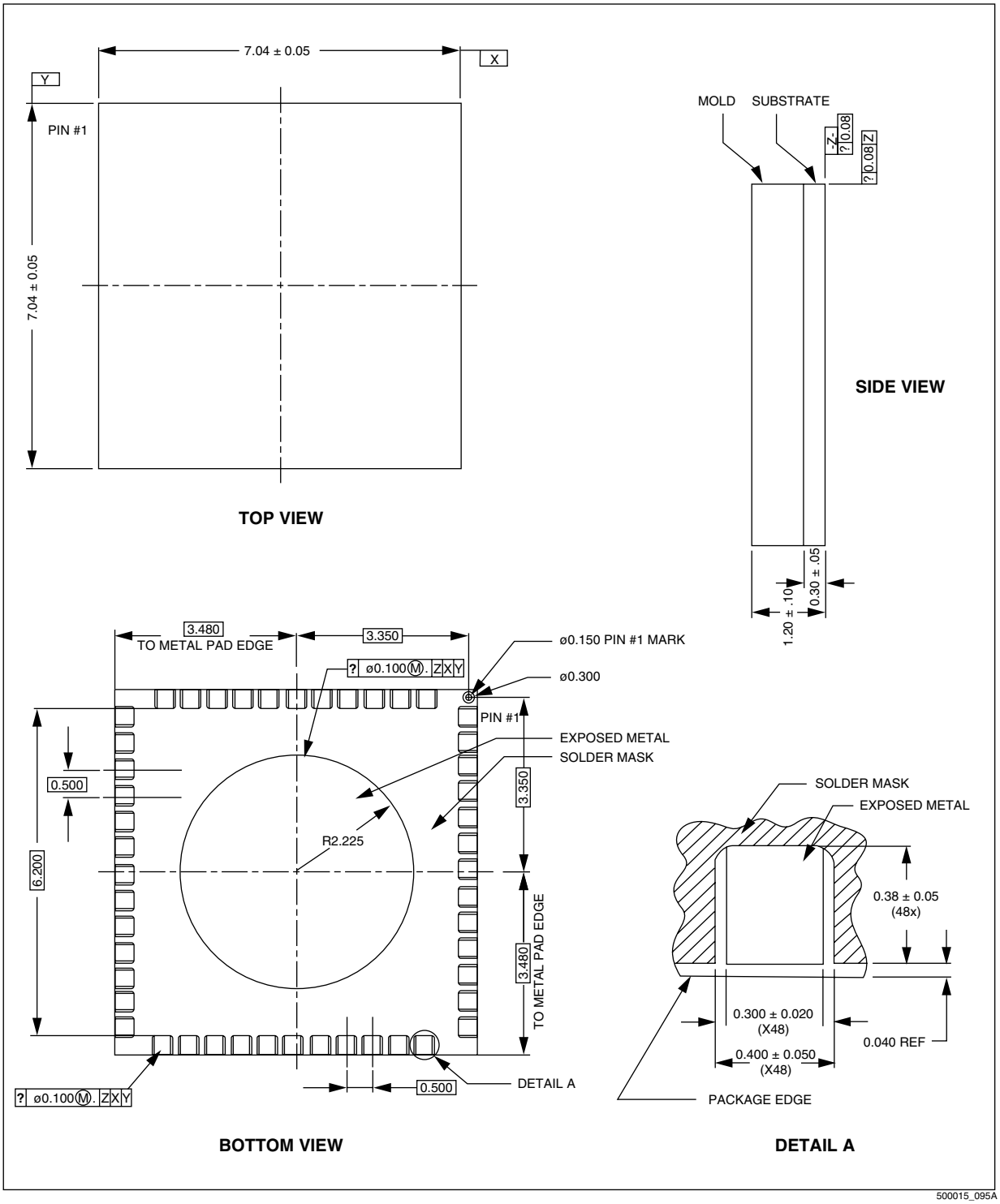
7.6.1.5 Solder Reflow Profile

Standard no-clean solder paste is generally recommended. If another type of flux is used, complete removal of flux residual may be necessary. Standard SMT reflow profiles can be used to surface mount the LGA package to the PWB. A range of recommended parameters for the SMT reflow profile are listed in Table 7-15. Additional soak time and slower preheating time may be required to improve the outgassing of solder paste during SMT reflow.

Table 7-15. Recommended SMT Reflow Profile

Description	Value
Preheat Slope (ambient to 120 °C)	1–2 °C/sec
Soak Slope (120–183 °C)	0.3–0.6 °C/sec
Time above reflow (> 183 °C)	50–80 sec
Peak Temperature	220 ± 5 °C
Cooling Rate	<6 °C/sec
Stencil	SS laser-cut, zero aperture reduction for LGA pads, 5-mil thickness
Test Package	5 × 5 LGA
PWB	FR4
Solder Paste	No-clean Sn63Pb37

Figure 7-13. Package Outline for 7 × 7 mm LGA



Appendix A: Acronyms and Abbreviations

ADC (A/D)	Analog-to-Digital Converter
AFE	Analog Front End
AIS	Alarm Indication Signal
API	Application Programming Interface
BER	Bit Error Rate
BGA	Ball Grid Array
BP	Bit Pump
BT	Bit Pump Transceiver
Channel Unit	HDSL Framer (name comes from HDSL1 Framer)
CRC-N	Cyclic Redundancy Check-N
CU	Channel Unit or HDSL Framer
DAC (D/A)	Digital-to-Analog Converter
DFE	Decision Feedback Equalizer
DIP	Dual In-Line Package
Downstream	From the HTU-C towards the HTU-R (includes regenerators)
DPLL	Digital Phase Lock Loop
DSL	Digital Subscriber Line
DSL Framer	ZipWireMulti DSL Framer Block
DSP	Digital Signal Processing
EC	Echo Canceller
EOC	Embedded Operations Channel
EVM	Evaluation Module
FEBE	Far End Block Error (the far end reported a CRC error)
FEXT	Far End Cross Talk
FFE	Feed Forward Equalizer

FIFO	First-In First-Out
FR	Framer
H2TU	HDSL2 Terminal Unit
HDLC	High-Level Data Link Controller
HDSL	High-Bit-Rate Digital Subscriber Line
HTU	HDSL Terminal Unit
HTU-C or COT or LTU	Central Office Terminal or Local Terminal Unit
HTU-R or RT or NTU	Remote Terminal or Network Terminal Unit
LED	Light Emitting Diode
LOS	Loss of Signal
NEXT	Near End Cross Talk
OOF	Out of Frame
P2MP	Point to Multipoint
PAM	Pulse Amplitude Modulation
PCM	Pulse Code Modulation
PLL	Phase Lock Loop
PRA	Primary Rate Access
PRBS	Pseudo-Random Bit Sequence
RAM	Random Access Memory
ROM	Read Only Memory
TCM	Trellis-Coded Modulation
TQFP	Thin Quad Flat Pack
Transceiver	ZipWireMulti DSP/Transceiver Block
UART	Universal Asynchronous Receive Transmit
UIP	User Interface Program
Upstream	From the HTU-R towards the HTU-C (includes regenerators)

MINDSPEED™

www.mindspeed.com

General Information:
U.S. and Canada: (800) 854-8099
International: (949) 483-6996
Headquarters - Newport Beach
4311 Jamboree Rd. P.O. Box C
Newport Beach, CA. 92658-8902