Rochester Electronics

Datasheet

PART NUMBER

PEF21628EV1X
PEF22628EV1X
PEF24628EV1X

Rochester Electronics Manufactured Components

Rochester branded components are manufactured using either die/wafers purchased from the original suppliers or Rochester wafers recreated from the original IP. All re-creations are done with the approval of the Original Component Manufacturer. (OCM)

Parts are tested using original factory test programs or Rochester developed test solutions to guarantee product meets or exceeds the OCM data sheet.

Quality Overview

- ISO-9001
- AS9120 certification
- Qualified Manufacturers List (QML) MIL-PRF-38535
    - Class Q Military
    - Class V Space Level

Qualified Suppliers List of Distributors (QSLD)

- Rochester is a critical supplier to DLA and meets all industry and DLA standards.

Rochester Electronics, LLC is committed to supplying products that satisfy customer expectations for quality and are equal to those originally supplied by industry manufacturers.

The original manufacturer's datasheet accompanying this document reflects the performance and specifications of the Rochester manufactured version of this device. Rochester Electronics guarantees the performance of its semiconductor products to the original OCM specifications. 'Typical' values are for reference purposes only. Certain minimum or maximum ratings may be based on product characterization, design, simulation, or sample testing.

ENVISIONING • EMPOWERING • EXCELLING

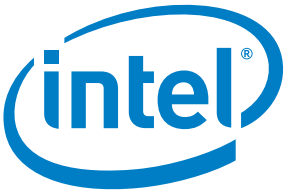
In 2020, MaxLinear acquired the
Connected Home Division business of Intel Corporation,
including the former Intel® product/s referenced in
the title of the attached material.

The MaxLinear logo will be added to the
attached material upon its next revision.

MaxLinear is now the manufacturer of this product, direct any questions and product support requests to your MaxLinear sales contact, MaxLinear Sales Representative or Distributor, or login to your myMxL account and create a new support ticket.

MaxLinear

Corporate Headquarters:
5966 La Place Court
Suite 100
Carlsbad, CA 92008
Tel.:+1 (760) 692-0711
Fax: +1 (760) 444-8598
www.maxlinear.com



Attention

The table below lists discontinued brand names that have now been replaced with **new Intel names**.

The discontinued brand names in this document originate from Lantiq Beteiligungs-GmbH & Co. KG or one of its predecessor companies, Infineon Technologies AG or Siemens AG.

Lantiq became part of the Intel Corporation on April 15th 2015.

Discontinued Name	New Intel Name	Category
SOCRATES™ EFM	Intel® SHDSL Chipset	Family
SOCRATES™-4e	PEF 24628	Device
SOCRATES™-2e	PEF 22628	Device
SOCRATES™-1e	PEF 21628	Device
SDFE-4	PEF 24624	Device
SBCX	Intel® ISDN Chipset for Terminals	Family
CONVERGATE™	Intel® Access Network Processor	Family

Legal Notice

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. Intel disclaims all warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

All information provided here is subject to change without notice. Intel may make changes to its test conditions and internal reliability goals at any time. Contact your Intel representative to obtain the latest Intel product specifications and road-maps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

SOCRATES™ EFM Family

SHDSL System Chip for EFM Application

SOCRATES™-4e (PEF 24628 E), Version 1.x
SOCRATES™-2e (PEF 22628 E), Version 1.x
SOCRATES™-1e (PEF 21628 E), Version 1.x

User's Manual

System Description
Revision 3.2, 2015-05-07

Confidential
Distribution with NDA by Marketing only

Edition 2015-05-07

Published by
Lantiq Beteiligungs-GmbH & Co.KG
Lilienthalstraße 15
85579 Neubiberg
Germany

Legal Disclaimer

The information given in this document is confidential.

SUCH INFORMATION SHALL IN NO EVENT BE REGARDED AS A GUARANTEE OF CONDITIONS OR CHARACTERISTICS. WITH RESPECT TO ANY VALUES STATED AS "TYPICAL" AS WELL AS EXAMPLES OR HINTS PROVIDED HEREIN, INCLUDING THOSE RELATED TO USE AND/OR IMPLEMENTATION OR APPLICATION OF COMPONENTS, LANTIQ BETEILIGUNGS-GMBH & CO.KG ("LANTIQ") HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. IN PARTICULAR, THOSE VALUES, EXAMPLES OR HINTS ARE NO WARRANTIES OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE LANTIQ COMPONENT AND ALWAYS HAVE TO BE VERIFIED IN THE TARGET APPLICATION.

The information in this document is subject to regular updates and/or corrections via errata sheet(s) or new revisions. It is only valid as updated and/or corrected. Once a new revision or errata sheet is available, it can be downloaded from Lantiq's Competence Center:

www.lantiq.com/contact-lantiq

Any user of this document must ensure that they only use the latest update- and/or corrected revision.

LANTIQ HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY, WITH RESPECT TO PREVIOUS REVISIONS OF THIS DOCUMENT ONCE AN UPDATE OR ERRATA SHEET IS AVAILABLE.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Lantiq office (**www.lantiq.com**).

Warning

Lantiq components must not be used in life-support devices or systems.

Life-support devices or systems are those systems intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Confidential

Revision History

Current:	**Revision 3.2, 2015-05-07**
Previous:	**Revision 3.1, 2011-06-18**

Page	Major changes since previous revision
178	Figure 108, PG-LBGA-324-8 (Plastic Low Profile Ball Grid Array Package) updated.
14,15,16	Several hints / explicit footnotes regarding FW release notes as reference for FW enabled feature set added (Posphy, RMII, HDLC-BZ, standalone mode / SPI, ...).
141-142	Figure 71, SS-SMII Cascading Mode in PHY Mode and description updated.

Registered Trademarks of Lantiq

AnyWAN™, CONVERGATE™, COSIC™, DSLTE™, DUALFALC™, DUSLIC™, ELIC™, EPIC™, FALC™, GEMINAX™, INCA™, ISAC™, IWORX™, OCTALFALC™, OCTAT™, QUADFALC™, SCOUT™, SEROCCO™, SICOFI™, SLIC™, SMINT™, SOCRATES™, VINAX™, VINETIC™, XWAY™

Other Trademarks

1-Wire® of Maxim Integrated. ARM® of ARM Ltd. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. eCosCentric®, eCos® and eCosPro® of eCosCentric Limited. EPCOS™ of Epcos AG. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IQfact™ and IQmax™ of LitePoint Corporation. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ispPAC® of LATTICE Semiconductor. MATLAB™ of MathWorks, Inc. NUCLEUS™ of Mentor Graphics Corporation. Linux® of Linus Torvalds. MIPS® of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO. SOLARIS™ of Sun Microsystems, Inc. Samtec® of Samtec Inc. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VxWorks™, WIND RIVER™ of WIND RIVER SYSTEMS.

All other trademarks are the property of their respective owners.

Last Trademarks Update 2014-10-17

Table of Contents

List of Figures

List of Tables

1 Introduction to Ethernet in the First Mile (EFM)

For bringing Ethernet packets directly from the central office side (CO) to the end user (CPE: Customer Premise Equipment) over voice-grade copper pairs, the Ethernet community standardized in IEEE 802.3-2004 the EFM PHYs 2BASE-TL referring to ITU G.shdsl.bis (192 kbit/s up to 5.7 Mbit/s) and 10PASS-TS referring to VDSL (2.5 - 100 Mbit/s). The PAF (PHY Aggregation Function) as standardized in IEEE 802.3-2004 allows channel bundling of EFM PHYs to either increase the data rate of one logical EFM link for a given loop length or to increase the maximum achievable loop length for a given data rate.

Lantiq's SOCRATES™ EFM family realizes 2BASE-TL PHYs including PAF as well as the EFM TC (64/65o).

For smooth migration Lantiq's SOCRATES™ EFM family is offering as well AAL5 functionality to adapt ATM cells to Ethernet packets and vice versa. This functionality enables i.e. ATM based DSLAMs to offer EFM based services.

1.1 BOM Optimization by Mounting Options

Lantiq's third generation SHDSL transceiver consists of 3 application optimized families. All family members are coming along in the same tiny package and are pin- and software-compatible[1)]. Hence the equipment BOM can be optimized by simple mounting options.

This document covers the SOCRATES™-4e/2e/1e devices of the SOCRATES™ EFM family, which is designed to serve Ethernet in the First Mile (EFM) gear. On top it allows a smooth transition between ATM and Ethernet world by offering interworking functionality (AAL5).

The SOCRATES™-4u/2u/1u devices of the SOCRATES™ EFM family are optimized for packet and ATM transport. In case of TDM based equipment the SDFE-4/2/1 devices of the SDFE family offer most competitive advantages.

For further information about these families please contact your local Lantiq representative.

1.2 Differences between the SOCRATES™-4e, SOCRATES™-2e and SOCRATES™-1e

The SOCRATES™-4e, SOCRATES™-2e and SOCRATES™-1e are full pin- and software compatible. All functions and interfaces described in this document are applicable for the SOCRATES™-4e as well as for the SOCRATES™-2e and for the SOCRATES™-1e.

The SOCRATES™-4e supports all 4 channels while the SOCRATES™-2e supports channel 0 and 3 and the SOCRATES™-1e supports channel 0 only. Thus in bonding or Four-wire/M-pair application the SOCRATES™-2e only supports a PAF bonding group of 2 pairs or Four-wire mode. The SOCRATES™-1e only supports a bonding group of 1.

1.3 Overview

The SOCRATES™-4e/2e/1e are the first members of Lantiq's third generation SHDSL transceiver.

They are designed to fulfill the ITU G.shdsl.bis standard as well as the EFM standard. This means in best SOCRATES™ tradition fully integration of the complete feature set including EFM PAF and AAL5.

For further members of this generation for ATM and packet based applications (see **Chapter 1.1**), please contact your local Lantiq office.

The SOCRATES™ EFM family addresses CO and CPE applications. At the central office side **EFM services can be offered out of ATM based DSLAMs, IP-DSLAMs, Ethernet Switches, Add-Drop-Multiplexer and DLCs**. For a detailed description please refer to **Chapter 1.3.3**, **Typical Applications**.

Various types of equipment are as well supported at the CPE side. The range is from **low cost stand alone modems up to high end EFM and ATM based modems/IAD** (see **Chapter 1.3.3**).

1) Beyond this every family consists of 1, 2 and 4 channel devices.

Due to pin and software compatibility between the 3rd generation families various kinds of equipment can be addressed with the same PCB and software by simple mounting options. In other words with a minimum development effort you get the capability to offer a broad range of cost and application optimized equipment.

1.3.1 Features

This chapter comprises an overview of the feature set implemented in the SOCRATES™-4e/2e/1e.

Since the hardware resources are enabled by firmware, refer to the Firmware Release Notes for the supported feature set and potential restrictions.

General

- Ethernet in the First Mile (EFM) accdg. to IEEE 802.3-2004 and ITU-T G.998.2 (including Amendment 2 - Support of BACP protocol)
- SHDSL.bis, extended data rates up to 5,7 Mbit/s ITU-T G.991.2 (2004) Annex F, Annex G
- Data rates up to 15 Mbit/s (proprietary mode)
- Fully integrated one chip system solution for Ethernet, Packets, ATM and TDM transport over 4/2 SHDSL channels
- Power Supply:
  – Digital Part: + 1.5 V
  – AFE, LD/Pads: + 3.3 V
- Integrated Controller handles autonomously channel bundling (PAF, M-pair, four-wire)
- Digital inputs and outputs TTL compatible
- Due to enhanced and flexible power saving modes the typical power consumption per SHDSL channel is only 450 - 650 mW depending on the mode
- High Level API for fast development cycle

SHDSL PHY

- DSP, analog frontend and Line Driver for 4/2/1 channels fully integrated into 1 package
- Asymmetric PSDs according to ITU-T G.991.2 (2004) Annex A and B fully supported
- Transmitting up to 16.8 dBm line power
- Only passive external components required, no inductors
- External analog circuitry same as for SDFE V1.3/V2.x family
- Software configurable operating speed per channel at any payload bit rate between 192 kbit/s and 11400 kbit/s fulfilling and exceeding ITU-T G.991.2 (2004) with a single crystal as reference clock by the same external circuitry
- Highly sophisticated line coding of two, three, four, five or six bits per symbol TC PAM
- Typical transmission range on 26 AWG (0.4mm) cable with 13.5 dBm line Power is 15 kft at 2304 kbit/s payload bit rate in noise-free environment
- Synchronous or plesiochronous (bit stuffing) data transfer possible
- Integrated hybrid which adapts automatically to the actual line conditions
- Adaptive echo cancellation and equalization
- Cross Talk Cancellation for up to 7 self-next disturbers (V1.2 only)
- Line Probing according to ITU-T G.991.2 (2004)
- Full digital clock recovery
- Input jitter tolerance according to ITU-T G.823, ITU-T G.824 and ITU-T G736. Jitter transfer function is better than specified in ITU-T I.430/I.431
- Integrated RAM for the complete firmware (no reload required)
- In Compliance with:
  – ETSI SDSL (ETSI TS 101 524 V 1.2.1)
  – ETSI SDSL.bis (ETSI TS 101 524 V 1.2.2)
  – ITU G.shdsl (ITU-T G.991.2)

  - ITU G.shdls.bis (ITU-T G.991.2 (2004)) including Annex F, Annex G and Amendment 3
  - ITU G.hs (ITU-T G.994.1)
  - IEEE EFM (IEEE 802.3-2004)
  - ITU g.998.2 (g.BOND)
- Interoperable with all Lantiq SHDSL transceivers and other standard compliant SHDSL transceivers
- Integrated Bit Error Tester

Modes

- STU-C or STU-R
- Analog, digital and remote loop backs
- MAC and PHY mode (see **Packet and Ethernet Modes**)

TPS-TC Framing

- ATM according to ITU-T G.991.2 (2004)
- 64/65o for EFM according to IEEE 802.3-2004
- HDLC for Packet according to ITU-T G.991.2 (2004)
- Serial data according to ITU-T G.991.2 (2004)

Channel Bonding

- PAF functionality from 1 group of 4 pairs to 4 groups of 1 pair according to IEEE 802.3-2004 (EFM) fully integrated (memory and controller integrated, i.e. no Host controller interactions required)
- Support of BACP protocol according to G.998.2 Amendment 2 for negotiating PAF groups
- M-pair and Four-wire mode according to ITU-T G.991.2 (2004) fully integrated (neither external memory nor Host controller interactions required)

Interworking

- 4 x AAL5 according to ITU-T I.363.5 (one per SHDSL link) selectable for each direction (packet on system side and ATM on the local loop side, ATM on system side and packets on the local loop side, i.e. support of EFM out of ATM based DSLAMs)
- Support of RFC 2684, LLC encapsulation and VC-MUX (up to 8 VCs according to TR-068 of DSL Forum)

System Interfaces

- **System Interface 1**
  - UTOPIA L2 acc. to af-phy-0039.000, 8/16-bit, up to 50 MHz, slave, 5 address lines & 1 CLAV/ENB group, cell level handshake, multiplexed status polling, no support of back to back cell transfers
  - POS-PHY L2 acc. to PMC-971147 issue 5, 16-bit, up to 50 MHz, slave, 5 address lines & 1 CLAV/ENB group, packet level mode, multiplexed status [1)]polling
  - 1*MII acc. to IEEE 802.3 in MAC mode (for connecting Ethernet 10/100 Mbit/s Ethernet PHYs) and PHY mode (for connecting SOCRATES™-4e to MAC controller)
    Full-duplex and half-duplex Ethernet at 10/100 Mbit/s
  - 4*RMII according to RMII consortium[1)]
  - 4*SS-SMII
- **System Interface 2**
  - 1.544 MHz clock, 193 bits / 125 µs frame
  - 2.048 MHz clock, 256 bits / 125 µs frame
  - 2.312 MHz clock, 289 bits / 125 µs frame
  - 4.096 MHz clock, 512 bits / 125 µs frame

1) Refer to FW release notes with respect to the level of support of the interface modes

  - 8.192 MHz clock, 1024 bits / 125 µs frame
  - 16.384 MHz clock, 2048 bits / 125 µs frame
  - 20.800 MHz clock, 2600 bits / 125 µs frame (9 x TU-12 + Signalling)
  - Generic bit-serial mode
  - Frame start programmable in TDM bit granularity
- System Interface 1 and System Interface 2 can work in parallel

Microcontroller Interface

- 8-bit asynchronous Motorola mode
- 8-bit synchronous Motorola mode
- 8-bit Intel demux mode
- Serial Control Interface

HDLC Controller for Z-Bits

Refer to FW release notes with respect to the level of support of this feature.

- One HDLC controller for all SHDSL channels
- Programmable to each SHDSL channel (on the fly switchable between the channels)
- Programmable to each TDM position
- Bit or Byte oriented HDLC
  - Bit oriented HDLC according to ITU-T Q.921
  - Byte oriented HDLC according to ITU-T G.997.1, §6.2 and IETF RFC 1662

HDLC Controller for EOC per Channel

- Byte oriented HDLC according to ITU-T G.991.2 (2004)

Firmware

- Important: Hardware resources are enabled by firmware, refer to the Firmware Release Notes for the supported feature set and potential restrictions/limitations
- SHDSL startup handled autonomously without involvement of external Host controller
- All SHDSL performance primitives and line related performance parameters according to ITU-T G.991.2 (2004) maintained
- Parameters relevant for clause 57 of IEEE 802.3-2004 (EFM) maintained
- Passive mode according to clause 57 of IEEE 802.3-2004 supported
- MIB objects defined in RFC2665/RFC3635 and RFC2863 supported
- Clause 45 registers of IEEE 802.3-2004 handled autonomously without involvement of external Host controller
- Control of IEEE 802.3-2004 clause 22 registers of connected PHY (MAC mode) without involvement of external Host controller
- F4/F5 for ATM according to ITU I.610 (AIS, RDI, CC and LB) handled autonomously without involvement of external Host controller

Software

- Configuration and maintenance of the chip based on SNMP
- High sophisticated easy programmable API
- EOC startup and messaging
- Performance Monitoring

Stand alone Mode for Low Cost CPE

Refer to FW release notes with respect to the level of support of this feature.

- No external Host controller required

- Firmware and configuration download directly from a Flash
- SHDSL startup autonomously initiated

Table 1 Supported Standard of the SOCRATES™-4e

Block of the Reference Model	Name	Standard
I/F	UTOPIA L2	ATM Forum, af-phy-0039.000
I/F	POS-PHY L2	PMC-971147 issue 5
I/F	MII	IEEE 802.3
TPS-TC	EFM	IEEE 802.3-2004
TPS-TC	PAF	IEEE 802.3-2004
TPS-TC	PTM (64/65 octet based)	ITU-T G.991.2 (Annex E.15)
TPS-TC	PAF	ITU-T G.994.1 (2003), Amd.2 (06/2004)
TPS-TC	ATM TC	ITU-T G.991.2 (Annex E.9)
TPS-TC	HDLC TC	ITU-T G.991.2 (Annex E.11)
TPS-TC	TDM	ITU-T G.991.2 (Annex E.4 - E.8)
TPS-TC	Dual Bearer	ITU-T G.991.2 (Annex E.10)
TPS-TC	M- pair/Four-wire	ITU-T G.991.2 (2004)
TPS-TC	AAL5	ITU-T I.363.5
		RFC 2684
PMD	ETSI SDSL	ETSI TS 101 524 V 1.2.1
PMD	ETSI SDSL.bis	ETSI TS 101 524 V 1.2.2
PMD	ITU G.shdsl	ITU-T G.991.2
PMD	ITU G.shdsl.bis	ITU-T G.991.2 (2004)

1.3.2 Logic Symbol

Figure 1 shows the logic symbol of the SOCRATES™-4e.

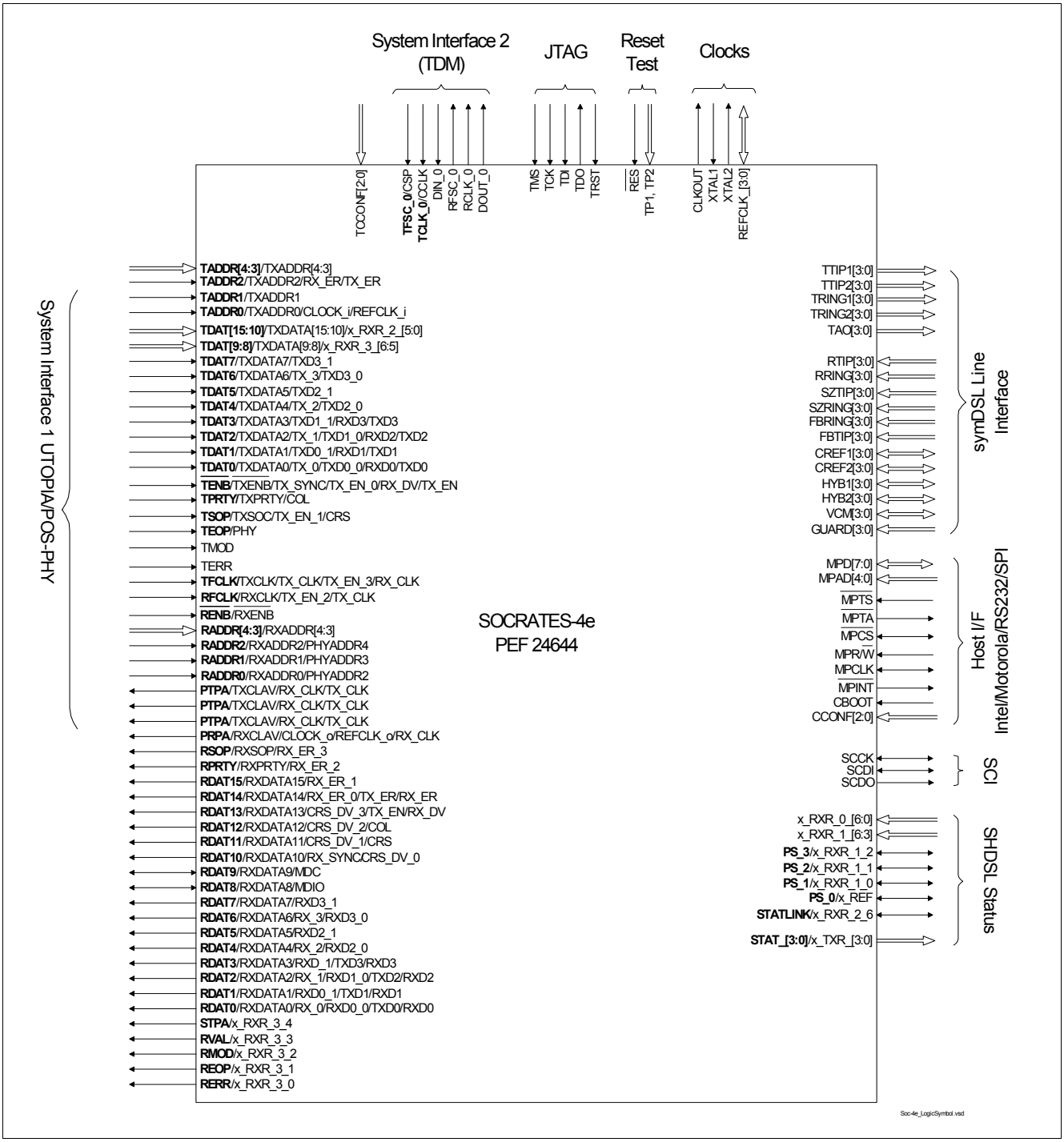


Figure 1 Logic Symbol of the SOCRATES™-4e

1.3.3 Typical Applications

This chapter depicts the typical applications served by the SOCRATES™ EFM family. CPE and COT applications of the SOCRATES™-4e are supported and illustrated in the following sub clauses.

Note: Host I/F mentioned in figures with xMII can also be MDIO

1.3.3.1 CPE Application

In this chapter the applications supported by the SOCRATES™ EFM family for CPE are described.

EFM Application

For EFM the MAC mode (behaves like an Ethernet MAC plus integrated EFM PHY) and PHY mode (behaves like an Ethernet EFM PHY) are supported (see **Chapter 3.3**).

In PHY mode the SOCRATES™-4e behaves from an external MAC perspective as an off the shelf Ethernet PHY, i.e. the MAC termination of the EFM link is realized by an external device, e.g. switch or router.

In MAC mode an Ethernet PHY can be directly connected to the SOCRATES™-4e. The SPI allows to connect a Flash memory to the SOCRATES™-4e to realize a stand alone application without any external host controller.

PHY Mode

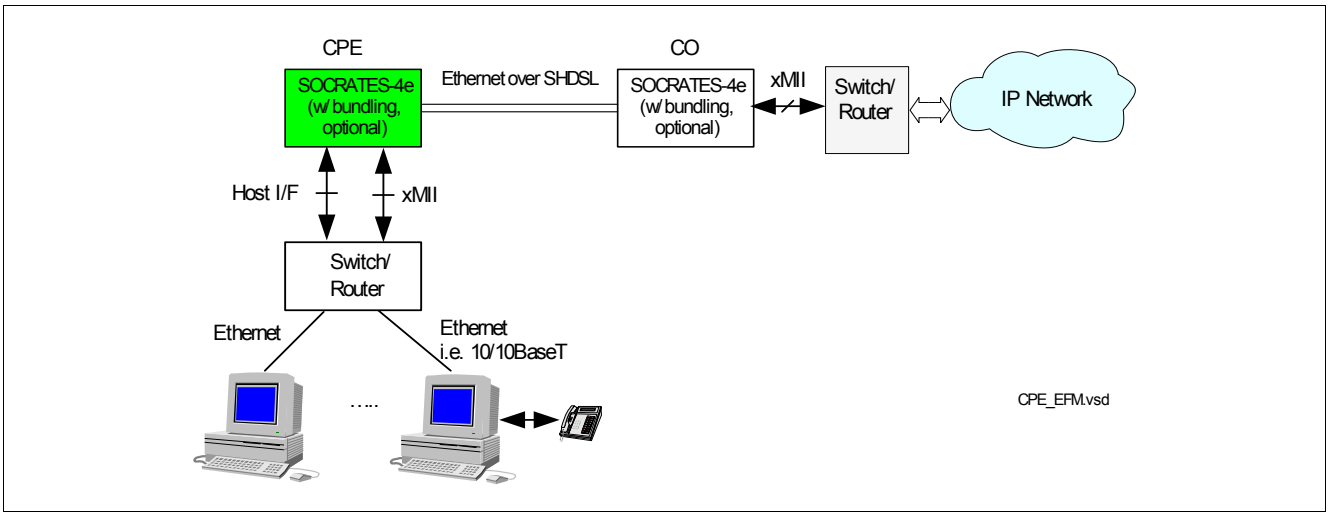


Figure 2 EFM Application, PHY Mode

Stand alone Mode (MAC Mode)

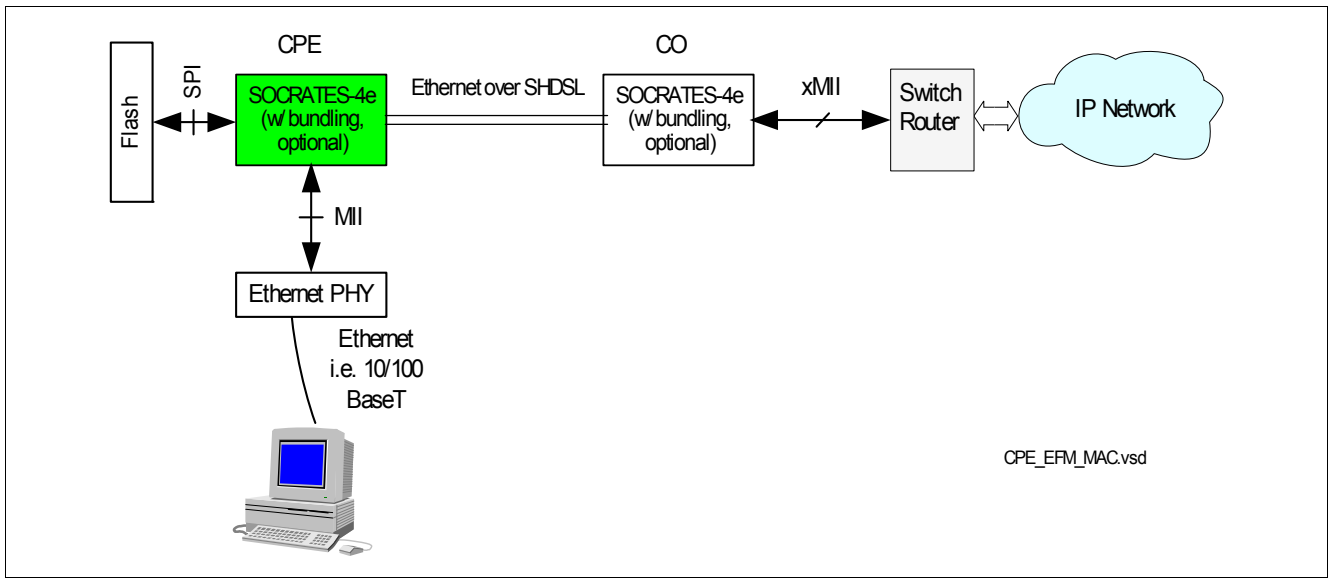


Figure 3 EFM Application, Stand alone Mode

Packet Application

This chapter presents applications utilizing packets in general transported over the SHDSL link.

As for EFM the MAC mode and PHY mode is supported for packet applications as well (see **Chapter 3.3**). Note that the only difference between the EFM applications and the packet applications in general is the use of the 64/65 octet TC in EFM application and the HDLC TC in packet applications.

PHY Mode

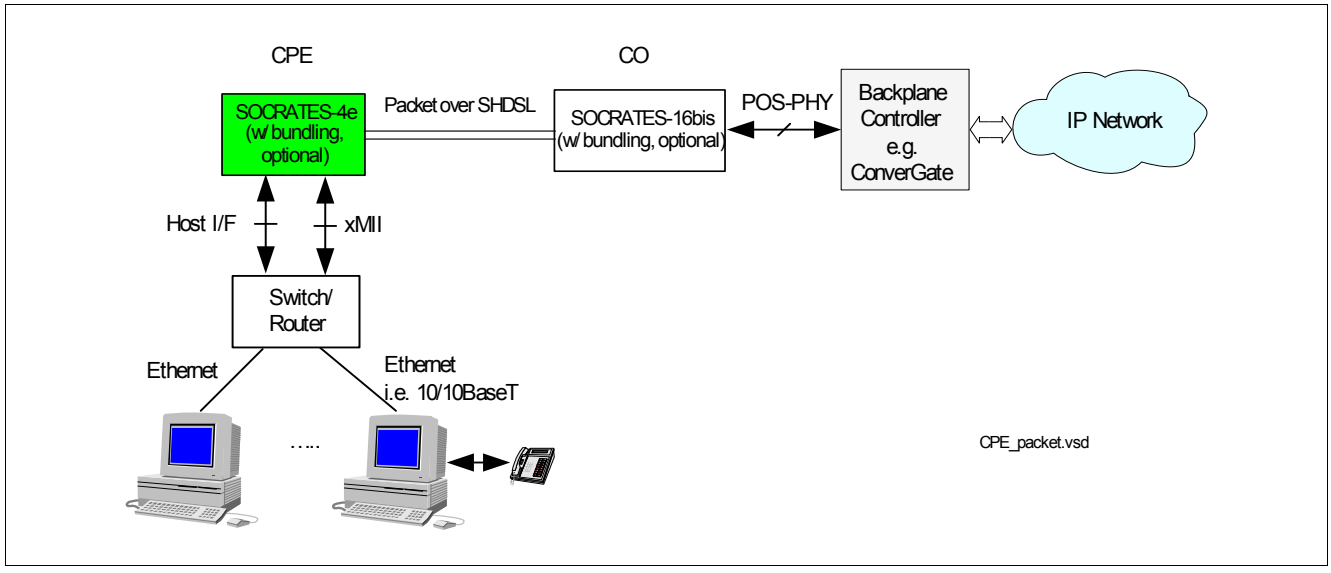


Figure 4 Packet Application, PHY Mode

Stand alone Mode (MAC Mode)

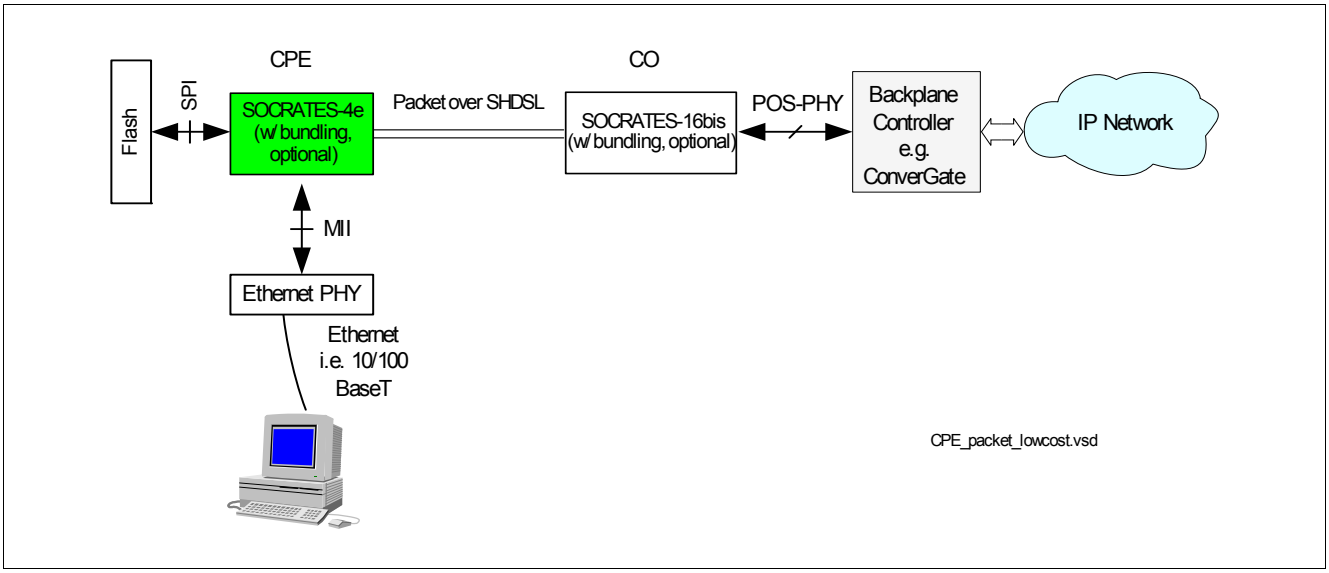


Figure 5 Packet Application, Stand alone Mode

ATM Application

This chapter presents an application utilizing ATM cells transported over the SHDSL link. Note that pure ATM applications can also be realized with the SOCRATES™-4u/2u/1u V3.1 devices of the SOCRATES™ EFM family.

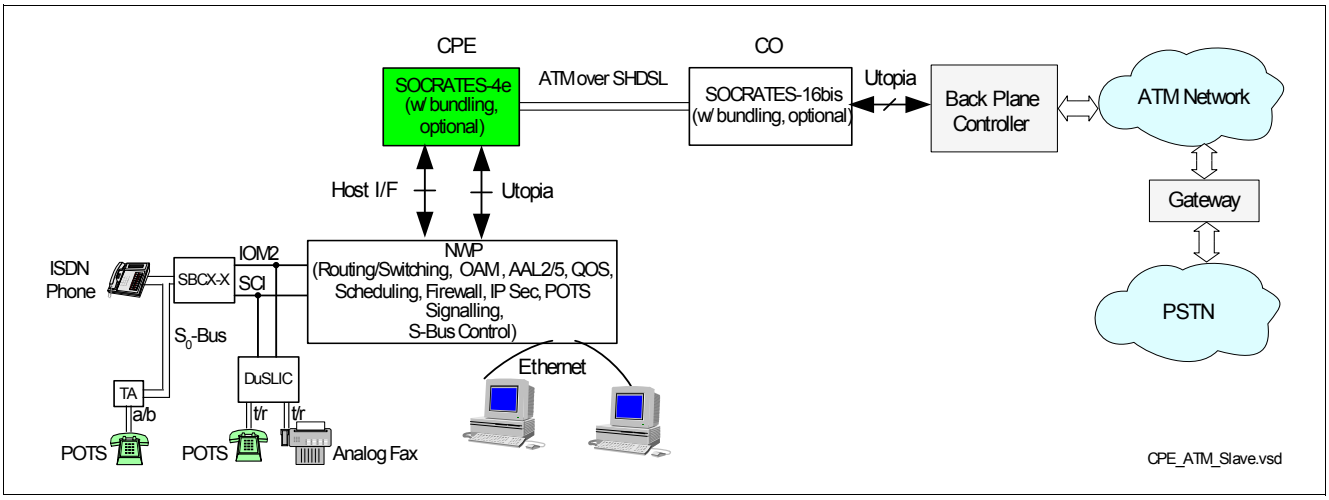


Figure 6 High End ATM CPE

Interworking Scenarios

In order to support the migration path from ATM cells on the SHDSL link to Ethernet frames on the SHDSL link, the SOCRATES™-4e supports the following interworking function:

- Ethernet CPE - ATM Cells on the SHDSL link

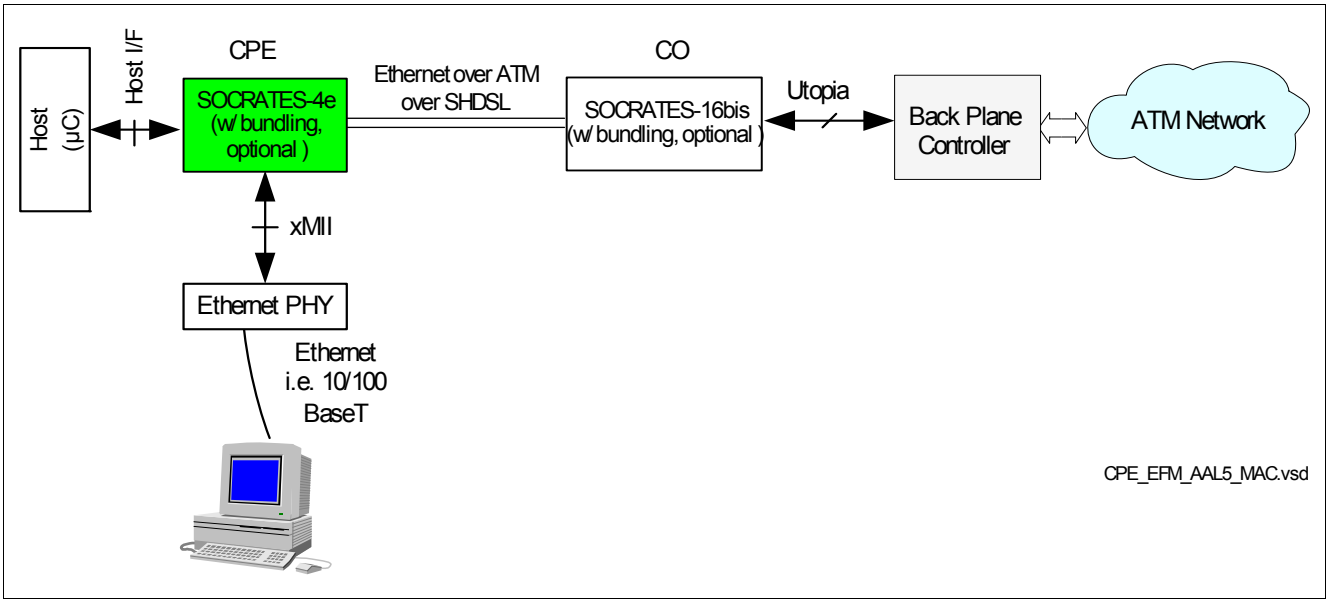


Figure 7 ATM Modem

Dual Bearer Modes

TDM data can be supported in all above applications in parallel to the packet/cell transport. The SOCRATES™-4e then operates in the so-called Dual Bearer Mode. TDM data will exchanged via the TDM interface. In dual bearer mode the entire SHDSL bandwidth will be divided between the 2 bearers in any ratio.

Note that the dual bearer mode is not standardized for all EFM applications according to IEEE 802.3-2004. Please refer to the SDFE family for TDM only applications.

1.3.3.2 CO Applications

In this chapter the applications supported by the SOCRATES™ EFM family for CO are described.

CO Ethernet Application

The following 2 applications demonstrate the SOCRATES™-4e in CO Ethernet environment. In both applications the SOCRATES™-4e acts in PHY mode (see **Chapter 3.3.1**). If the 64/65octet TC is chosen the SOCRATES™-4e behaves like a EFM PHY. Alternatively, the HDLC TC can be chosen as well.

Ethernet, CO Switching/Routing Application

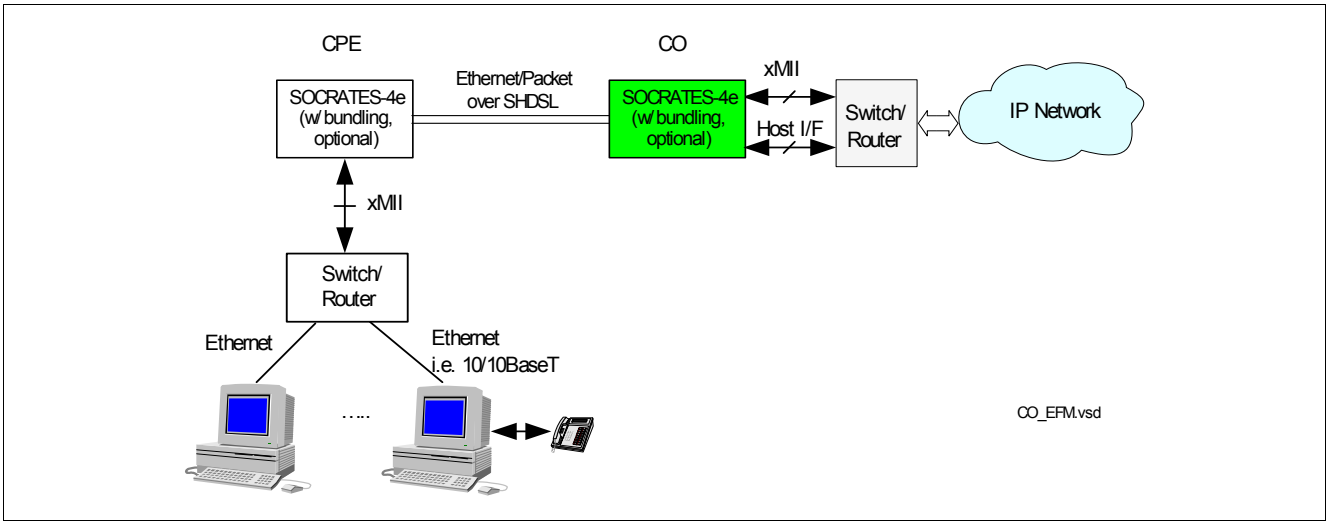


Figure 8 CO Ethernet Switching/Routing

Ethernet, IP-DSLAM Application

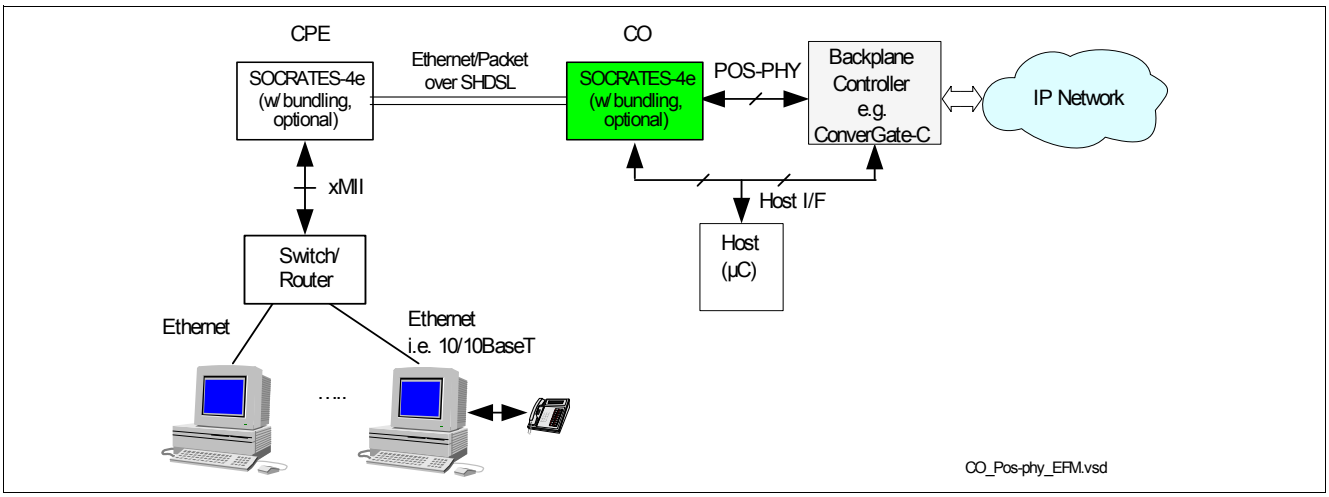


Figure 9 IP-DSLAM

EFM/Ethernet out of ATM based DSLAM

The following 2 applications demonstrate the SOCRATES™-4e transporting Ethernet out of DSLAMs. The SOCRATES™-4e offers the UTOPIA or POS-PHY interface as the system interface for this application.

In both applications the SOCRATES™-4e adapts Ethernet frames to ATM cells and vice versa according to ITU-T I.363.5 and RFC 2684. As in previous examples the only difference between the 2 applications is the use of the TC (64/65octet vs HDLC).

Ethernet out of ATM based DSLAM, 64/65o TC (EFM)

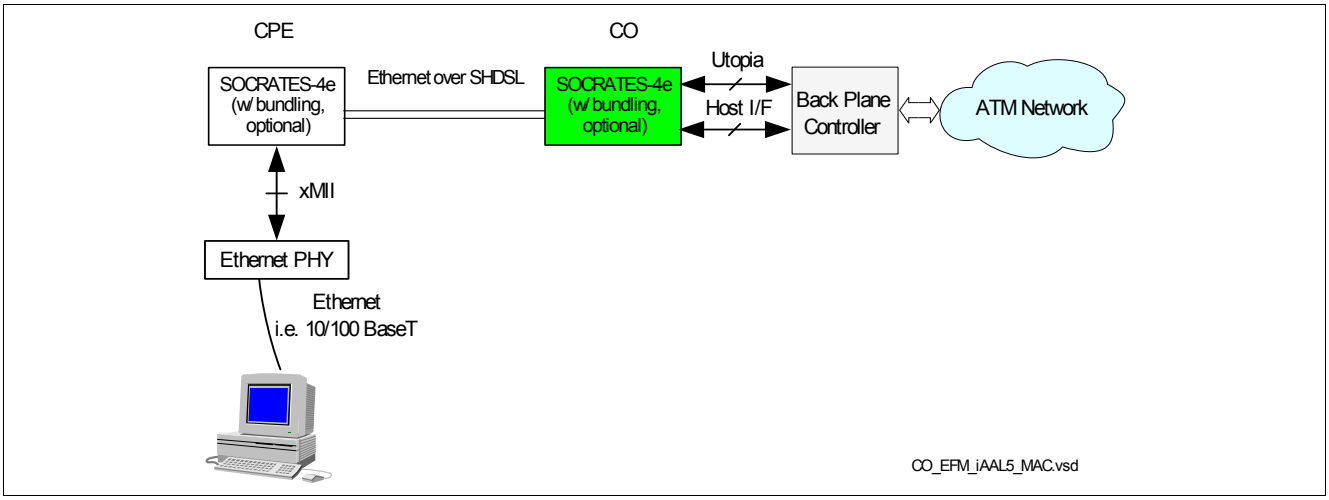


Figure 10 ATM based DLAM for EFM with UTOPIA I/F

Ethernet out of ATM based DSLAM, PTM TC

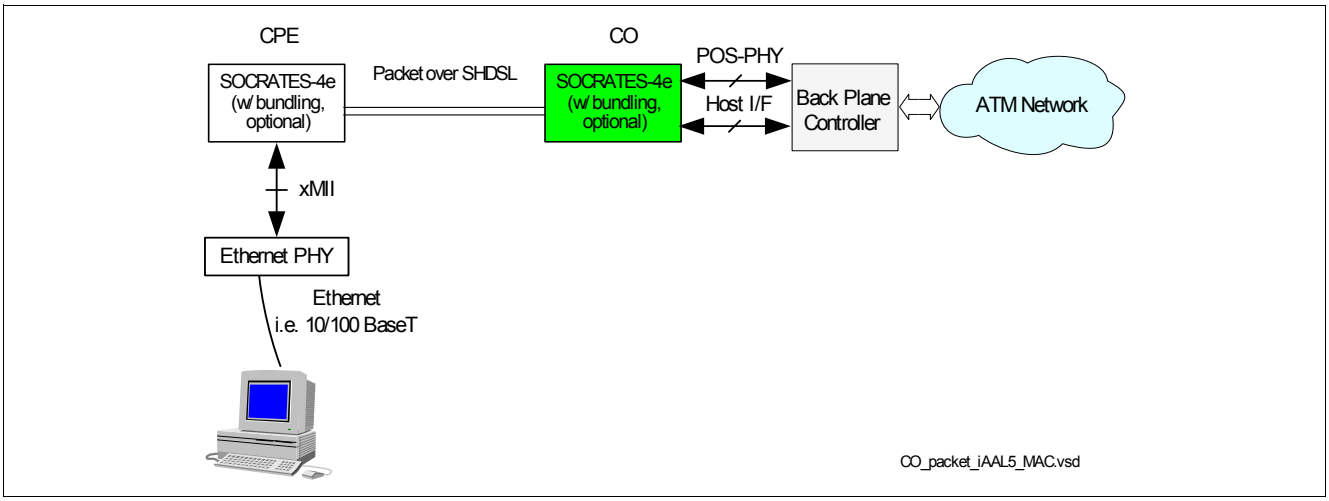


Figure 11 ATM based DLAM for Ethernet with POS-PHY I/F

CO ATM Application

The following applications shows the SOCRATES™-4e transporting pure ATM data. Note that pure ATM applications can also be realized with the SOCRATES™-4u/2u/1u devices of the SOCRATES™ EFM family in a cost optimized way.

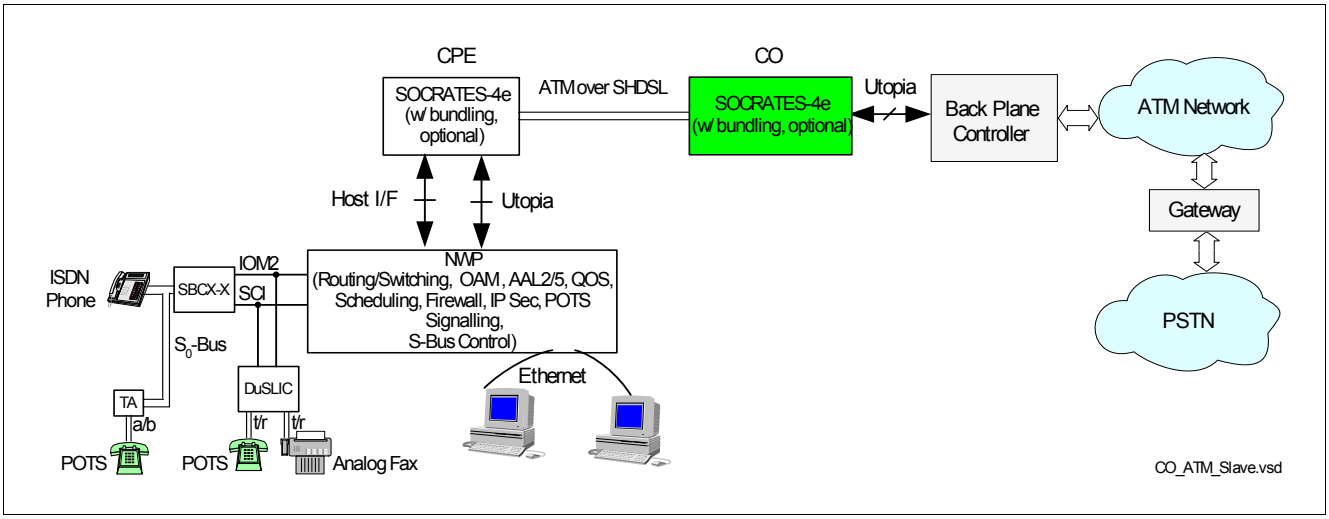


Figure 12 CO ATM Application

Dual Bearer Modes

TDM data can be supported in all above applications in parallel to the packet/cell transport. The SOCRATES™-4e then operates in the so-called Dual Bearer Mode. TDM data will be exchanged via the TDM interface. In dual bearer mode the entire SHDSL bandwidth will be divided between the 2 bearers in any ratio. Note that the dual bearer mode is not standardized for all EFM applications according to IEEE 802.3-2004. Please refer to the SDFE family for TDM only applications.

1.4 Coding of the Firmware Releases

The SOCRATES™-4e needs a dedicated firmware. The coding of the firmware releases is done in the following way:

Release a.b-c.d.e means

- This firmware release can be downloaded to the SOCRATES™-4e HW Va.b
  (a.b = HW ID)
- This firmware release is step c of the planned firmware steps
  (c = FW major feature package ID)
- This firmware release is variant d of step c with full or reduced functionality of step c
  (d = FW feature package ID)
- This firmware release has been updated (e - 1) times
  (e = FW development status ID)

As an example release 1.1-1.2.3 would mean that the firmware is developed for the HW version 1.1 and that this is the 1st functionality step with the reduced functionality variant 2, 2 times updated.

1.5 Terms and Definitions

This chapter gives an overview of the used terms.

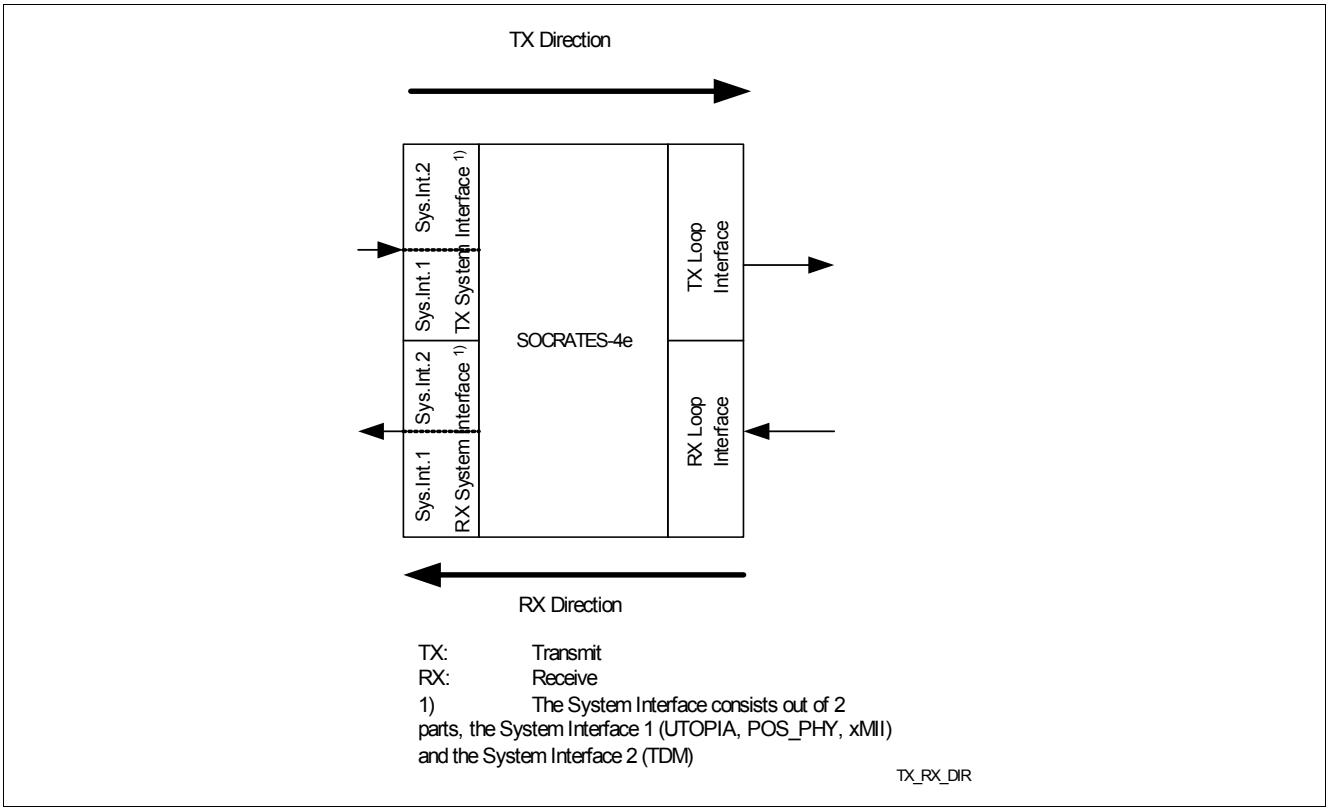


Figure 13 Terminology for Direction

TX stands for Transmit Direction, RX for RX direction. The data flow from the System Interface to the Loop Interface is called TX direction whereas the data flow from the Loop Interface to the System Interface is called RX direction.

The System Interface itself consists out of 2 parts. The System Interface 1 carries parallel data and can be configured as UTOPIA, POS_PHY or xMII interface. The System Interface 2 carries serial data and will be operated in TDM mode.

Other Used Terms

A packet is always an Ethernet Frame according to IEEE 802.3-2004.

The term fragment is used twice. When used at the System Interface 1 it refers to a part of an Ethernet Frame being carried in portion over the POS-PHY interface. When used with the Bonding/Aggregation Function it refers to a part of an Ethernet Frame added by some overhead information needed for the Aggregation Function.

2 External Signals

The SOCRATES™-4e can be operated in different modi as shown in **Chapter 1.3.3**. Different interfaces for carrying payload data (accessed via System Interface 1) will be used in these different applications. The different interface signals at System Interface 2 are multiplexed. In order to allow a correlation between the different applications and the pinning/ball diagram the ball diagrams together with ball definitions are shown for the different applications. Note that the differences only apply to the System Interface 1.

2.1 Ball Diagram

This chapter shows the ball diagrams when the SOCRATES™-4e works in PHY mode (System Interface 1 in xMII PHY Mode), in MAC mode (System Interface 1 in xMII MAC mode) and in packet/cell applications (System Interface 1 in UTOPIA/POS-PHY mode.

2.1.1 Ball Diagram, UTOPIA/POS-PHY Mode

	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	RFCLK	TDAT15	RDAT8	TERR	TCK	TRST	TMS	MPD7	x_RXR_0_6	VSSA	CREF2_2	CREF1_2	VSSA	CREF2_3	CREF1_3	VSSA	VDDA	VSSA	A
B	PTPA	TFCLK	TPRTY	TMOD	TDO	RPRTY	TDI	MPD6	x_RXR_0_5	VSSA	GUARD_3	TAO_3	FBTIP_3	HYB1_3	RTIP_3	SZTIP_3	TTIP2_3	TTIP1_3	B
C	PRPA	TADDR0	STPA	REFCLK_3	RVAL	MPD2	RADDR1	MPD5	x_RXR_0_4	VSSA	GUARD_2	TAO_2	FBRING_3	HYB2_3	RRING_3	SZRING_3	TRING2_3	TRING1_3	C
D	TSOP	REFCLK_2	TADDR1	MPD0	MPD1	MPD3	RADDR0	MPD4	x_RXR_0_3	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_3	D
E	TENB	RDAT10	TADDR2	TDAT14	TDAT13	TDAT12	TDAT11	TDAT10	x_RXR_0_2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	E
F	TDAT0	RDAT0	TADDR3	VSSD	VDDD	TDAT9	VDDP	VSSD	x_RXR_0_1	VSSA	VDDA	VSSA	FBTIP_2	HYB1_2	RTIP_2	SZTIP_2	TTIP2_2	TTIP1_2	F
G	TDAT1	RDAT1	TADDR4	VSSD	VDDD	TDAT8	VDDP	VSSD	x_RXR_0_0	VSSA	VDDA	VSSA	FBRING_2	HYB2_2	RRING_2	SZRING_2	TRING2_2	TRING1_2	G
H	TDAT2	RDAT2	RENB	VSSD	VDDD	STAT_3	VDDP	VSSD	TCCONF0	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_2	H
J	TDAT3	RDAT3	RSOP	VSSD	VDDD	STAT_2	VDDP	VSSD	TCCONF1	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	J
K	TDAT4	RDAT4	RDAT15	VSSD	VDDD	STAT_1	VDDP	VSSD	TCCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	K
L	TDAT5	RDAT5	RDAT14	VSSD	VDDD	STAT_0	VDDP	VSSD	CBOOT	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_1	L
M	TDAT6	RDAT6	RDAT13	VSSD	VDDD	STATLINK	VDDP	VSSD	CCONF0	VSSA	VDDA	VSSA	FBRING_1	HYB2_1	RRING_1	SZRING_1	TRING2_1	TRING1_1	M
N	TDAT7	RDAT7	RDAT12	VSSD	VDDD	x_RXR_1_3	VDDP	VSSD	CCONF1	VSSA	VDDA	VSSA	FBTIP_1	HYB1_1	RTIP_1	SZTIP_1	TTIP2_1	TTIP1_1	N
P	SCDO	RDAT11	PS_0	PS_1	PS_2	PS_3	x_RXR_1_4	x_RXR_1_5	CCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	P
R	SCCK	MPAD0	MPAD1	MPAD2	MPAD3	MPAD4	MPTA	MPCS	x_RXR_1_6	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_0	R
T	SCDI	RDAT9	RFSC_0	TEOP	RMOD	MPCLK	RADDR3	MPR	CLKOUT	VSSA	GUARD_1	TAO_1	FBRING_0	HYB2_0	RRING_0	SZRING_0	TRING2_0	TRING1_0	T
U	DIN_0	REFCLK_0	RCLK_0	REFCLK_1	TP1	RADDR2	TP2	MPTS	XTAL1	VSSA	GUARD_0	TAO_0	FBTIP_0	HYB1_0	RTIP_0	SZTIP_0	TTIP2_0	TTIP1_0	U
V	DOUT_0	TFSC_0	TCLK_0	REOP	RERR	RES	RADDR4	MPINT	XTAL2	VSSA	CREF2_1	CREF1_1	VSSA	CREF2_0	CREF1_0	VSSA	VDDA	VSSA	V
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 14 Ball Diagram PG-LBGA-324-2 (Ball View), UTOPIA/POS-PHY Mode

2.1.2 Ball Diagram, PHY Mode

	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	TX_EN_2	x_RXR_2_5	MDIO	PU	TCK	TRST	TMS	MPD7	x_RXR_0_6	VSSA	CREF2_2	CREF1_2	VSSA	CREF2_3	CREF1_3	VSSA	VDDA	VSSA	A
B	TX_CLK	TX_EN_3	PU	PU	TDO	RX_ER_2	TDI	MPD6	x_RXR_0_5	VSSA	GUARD_3	TAO_3	FBTIP_3	HYB1_3	RTIP_3	SZTIP_3	TTIP2_3	TTIP1_3	B
C	REFCLK_o	REFCLK_i	x_RXR_3_4	REFCLK_3	x_RXR_3_3	MPD2	PHYADDR3	MPD5	x_RXR_0_4	VSSA	GUARD_2	TAO_2	FBRING_3	HYB2_3	RRING_3	SZRING_3	TRING2_3	TRING1_3	C
D	TX_EN_1	REFCLK_2	MDIOSLAV	MPD0	MPD1	MPD3	PHYADDR2	MPD4	x_RXR_0_3	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_3	D
E	TX_EN_0	CRS_DV_0	TX_ER	x_RXR_2_4	x_RXR_2_3	x_RXR_2_2	x_RXR_2_1	x_RXR_2_0	x_RXR_0_2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	E
F	TXD0_0	RXD0_0	PU	VSSD	VDDD	x_RXR_3_6	VDDP	VSSD	x_RXR_0_1	VSSA	VDDA	VSSA	FBTIP_2	HYB1_2	RTIP_2	SZTIP_2	TTIP2_2	TTIP1_2	F
G	TXD0_1	RXD0_1	PU	VSSD	VDDD	x_RXR_3_5	VDDP	VSSD	x_RXR_0_0	VSSA	VDDA	VSSA	FBRING_2	HYB2_2	RRING_2	SZRING_2	TRING2_2	TRING1_2	G
H	TXD1_0	RXD1_0	PU	VSSD	VDDD	STAT_3	VDDP	VSSD	TCCONF0	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_2	H
J	TXD1_1	RXD1_1	RX_ER_3	VSSD	VDDD	STAT_2	VDDP	VSSD	TCCONF1	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	J
K	TXD2_0	RXD2_0	RX_ER_1	VSSD	VDDD	STAT_1	VDDP	VSSD	TCCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	K
L	TXD2_1	RXD2_1	RX_ER_0	VSSD	VDDD	STAT_0	VDDP	VSSD	CBOOT	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_1	L
M	TXD3_0	RXD3_0	CRS_DV_3	VSSD	VDDD	STATLINK	VDDP	VSSD	CCONF0	VSSA	VDDA	VSSA	FBRING_1	HYB2_1	RRING_1	SZRING_1	TRING2_1	TRING1_1	M
N	TXD3_1	RXD3_1	CRS_DV_2	VSSD	VDDD	x_RXR_1_3	VDDP	VSSD	CCONF1	VSSA	VDDA	VSSA	FBTIP_1	HYB1_1	RTIP_1	SZTIP_1	TTIP2_1	TTIP1_1	N
P	SCDO	CRS_DV_1	PS_0	PS_1	PS_2	PS_3	x_RXR_1_4	x_RXR_1_5	CCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	P
R	SCCK	MPAD0	MPAD1	MPAD2	MPAD3	MPAD4	MPTA	MPCS	x_RXR_1_6	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_0	R
T	SCDI	MDC	RFSC_0	PHY	x_RXR_3_2	MPCLK	PU	MPR	CLKOUT	VSSA	GUARD_1	TAO_1	FBRING_0	HYB2_0	RRING_0	SZRING_0	TRING2_0	TRING1_0	T
U	DIN_0	REFCLK_0	RCLK_0	REFCLK_1	TP1	PHYADDR4	TP2	MPTS	XTAL1	VSSA	GUARD_0	TAO_0	FBTIP_0	HYB1_0	RTIP_0	SZTIP_0	TTIP2_0	TTIP1_0	U
V	DOUT_0	TFSC_0	PCLK_0	x_RXR_3_1	x_RXR_3_0	RES	SYNC_i	MPINT	XTAL2	VSSA	CREF2_1	CREF1_1	VSSA	CREF2_0	CREF1_0	VSSA	VDDA	VSSA	V
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 15 Ball Diagram PG-LBGA-324-2 (Ball View), PHY Mode

2.1.3 Ball Diagram, MAC Mode

	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
A	CRS_DV_2	x_RXR_2_5	MDIO	PU	TCK	TRST	TMS	MPD7	x_RXR_0_6	VSSA	CREF2_2	CREF1_2	VSSA	CREF2_3	CREF1_3	VSSA	VDDA	VSSA	A
B	TX_CLK	CRS_DV_3	COL	PU	TDO	PU	TDI	MPD6	x_RXR_0_5	VSSA	GUARD_3	TAO_3	FBTIP_3	HYB1_3	RTIP_3	SZTIP_3	TTIP2_3	TTIP1_3	B
C	REFCLK_o	REFCLK_i	x_RXR_3_4	REFCLK_3	x_RXR_3_3	MPD2	PHYADDR3	MPD5	x_RXR_0_4	VSSA	GUARD_2	TAO_2	FBRING_3	HYB2_3	RRING_3	SZRING_3	TRING2_3	TRING1_3	C
D	CRS_DV_1	REFCLK_2	MDIOSLAV	MPD0	MPD1	MPD3	PHYADDR2	MPD4	x_RXR_0_3	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_3	D
E	CRS_DV_0	TX_EN_0	RX_ER_0	x_RXR_2_4	x_RXR_2_3	x_RXR_2_2	x_RXR_2_1	x_RXR_2_0	x_RXR_0_2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	E
F	RXD0_0	TXD0_0	RX_ER_1	VSSD	VDDD	x_RXR_3_6	VDDP	VSSD	x_RXR_0_1	VSSA	VDDA	VSSA	FBTIP_2	HYB1_2	RTIP_2	SZTIP_2	TTIP2_2	TTIP1_2	F
G	RXD0_1	TXD0_1	RX_ER_2	VSSD	VDDD	x_RXR_3_5	VDDP	VSSD	x_RXR_0_0	VSSA	VDDA	VSSA	FBRING_2	HYB2_2	RRING_2	SZRING_2	TRING2_2	TRING1_2	G
H	RXD1_0	TXD1_0	RX_ER_3	VSSD	VDDD	STAT_3	VDDP	VSSD	TCCONF0	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_2	H
J	RXD1_1	TXD1_1	SYNC_o	VSSD	VDDD	STAT_2	VDDP	VSSD	TCCONF1	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	J
K	RXD2_0	TXD2_0	PU	VSSD	VDDD	STAT_1	VDDP	VSSD	TCCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	K
L	RXD2_1	TXD2_1	TX_ER	VSSD	VDDD	STAT_0	VDDP	VSSD	CBOOT	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_1	L
M	RXD3_0	TXD3_0	TX_EN_3	VSSD	VDDD	STATLINK	VDDP	VSSD	CCONF0	VSSA	VDDA	VSSA	FBRING_1	HYB2_1	RRING_1	SZRING_1	TRING2_1	TRING1_1	M
N	RXD3_1	TXD3_1	TX_EN_2	VSSD	VDDD	x_RXR_1_3	VDDP	VSSD	CCONF1	VSSA	VDDA	VSSA	FBTIP_1	HYB1_1	RTIP_1	SZTIP_1	TTIP2_1	TTIP1_1	N
P	SCDO	TX_EN_1	PS_0	PS_1	PS_2	PS_3	x_RXR_1_4	x_RXR_1_5	CCONF2	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	P
R	SCCK	MPAD0	MPAD1	MPAD2	MPAD3	MPAD4	MPTA	MPCS	x_RXR_1_6	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VSSA	VDDA	VCM_0	R
T	SCDI	MDC	RFSC_0	PHY	x_RXR_3_2	MPCLK	PU	MPR	CLKOUT	VSSA	GUARD_1	TAO_1	FBRING_0	HYB2_0	RRING_0	SZRING_0	TRING2_0	TRING1_0	T
U	DIN_0	REFCLK_0	RCLK_0	REFCLK_1	TP1	PHYADDR4	TP2	MPTS	XTAL1	VSSA	GUARD_0	TAO_0	FBTIP_0	HYB1_0	RTIP_0	SZTIP_0	TTIP2_0	TTIP1_0	U
V	DOUT_0	TFSC_0	PCLK_0	x_RXR_3_1	x_RXR_3_0	RES	SYNC_i	MPINT	XTAL2	VSSA	CREF2_1	CREF1_1	VSSA	CREF2_0	CREF1_0	VSSA	VDDA	VSSA	V
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 16 Ball Diagram PG-LBGA-324-2 (Ball View), MAC Mode

2.2 Ball Definitions and Functions

Bold names in the following tables are names of the pin in package. Note that the System Interface 1 section contains different descriptions according to the ball diagrams shown in **Figure 14**, **Figure 15** and **Figure 16**.

2.2.1 General

Table 2 I/O Signals, Reset and Test

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
V13	**$\overline{\text{RES}}$**	I		**Reset** Active Low
U14	**TP1**	I		**Test Pin 1** Connect to ’0’ when using XTAL frequency of 20.48 MHz Connect to ’1’ when using XTAL frequency of 40.96 MHz
U12	**TP2**	I		**Test Pin 2** Connect to ’0’ for both XTAL frequencies of 20.48 MHz and 40.96 MHz *Note: Has to be connected to ’0’ when the Boundary Scan is performed*

Table 3 I/O Signals, SHDSL Reference Clocks

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C15	REFCLK_3	I/O	PP	**Reference Clock, SHDSL Port 3** Buffer: C
D17	REFCLK_2	I/O	PP	**Reference Clock, SHDSL Port 2** Buffer: C
U15	REFCLK_1	I/O	PP	**Reference Clock, SHDSL Port 1** Buffer: C
U17	REFCLK_0	I/O	PP	**Reference Clock, SHDSL Port 0** Buffer: C

2.2.2 Configuration

Table 4 I/O Signals, Host Interface Mode Settings

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
P10	CCONF2	I		**CCONF[2:0]** Configuration of Host Interface. 000_B **Intel Demux**, Intel Demux 001_B **Motorola Async**, Motorola asynchronous 010_B **Motorola Sync**, Motorola synchronous 011_B **RS232**, RS232 and FLASH 100_B **SCI**, Serial Control Interface
N10	CCONF1			
M10	CCONF0			
L10	CBOOT			**Configuration of Boot Mode** 0_B **Boot from ROM**, boot from internal ROM, wait for firmware download via Host interface, need to be selected in case of RS232/SPI used as Host Interface 1_B **Boot from RAM**, firmware download via Host interface, booting from internal RAM initiated by host

2.2.3 Status

The status group offers three different modes. Mode selection has to be done by a dedicated message.

In normal mode (reset behavior) the pins of this group does not contain valid information. The second mode is the status mode. Since a distinction has to be made between the STU-C (CO side) and the STU-R (CPE), a status per side is offered (STAT_STU_R for status in a CPE application; STAT_STU_C for status in CO application). For the power status inputs in the CPE application the following coding has to be applied: 1 = Power status UP, 0 = Power Status FAILURE. The same values are driven on the PS pins at the CO side. For STATLINK the following coding applies: 0 = status of the link is down, 1 = status of the link is up, data transfer is possible. The STAT_x pins refer to the status of the physical layer, the following coding applies: 0 = physical layer down, 1 = physical layer up.

The third mode is used in crosstalk application. This mode has to be enabled as well by a message. Note that enabling the crosstalk mode uses also some pins of the UTOPIA/POS-PHY interface, therefor the UTOPIA/POS-PHY interface is not available in crosstalk mode.

Other Not used inputs without an internal pull up resistor have to be connected to ’1’.

Table 5 I/O Signals, SHDSL Status

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
A10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_6	I		**xTALK: Rx Reference Signal[6], Channel 0**
	-	I	PU1	**Normal: Not used**
B10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_5	I		**xTALK: Rx Reference Signal[5], Channel 0**
	-	I	PU1	**Normal: Not used**

Table 5 I/O Signals, SHDSL Status (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_4	I		**xTALK: Rx Reference Signal[4], Channel 0**
	-	I	PU1	**Normal: Not used**
D10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_3	I		**xTALK: Rx Reference Signal[3], Channel 0**
	-	I	PU1	**Normal: Not used**
E10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_2	I		**xTALK: Rx Reference Signal[2], Channel 0**
	-	I	PU1	**Normal: Not used**
F10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_1	I		**xTALK: Rx Reference Signal[1], Channel 0**
	-	I	PU1	**Normal: Not used**
G10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_0_0	I		**xTALK: Rx Reference Signal[0], Channel 0**
	-	I	PU1	**Normal: Not used**
R10	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_1_6	I		**xTALK: Rx Reference Signal[6], Channel 1**
	-	I	PU1	**Normal: Not used**
P11	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_1_5	I		**xTALK: Rx Reference Signal[5], Channel 1**
	-	NU		**Normal: Not used**
P12	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_1_4	I		**xTALK: Rx Reference Signal[4], Channel 1**
	-	I	PU1	**Normal: Not used**
N13	-	I	PU1	**STAT_STU_R: Not used**
	-	I	PU1	**STAT_STU_C: Not used**
	x_RXR_1_3	I		**xTALK: Rx Reference Signal[3], Channel 1**
	-	NU		**Normal: Not used**

Table 5 I/O Signals, SHDSL Status (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
P13	**PS_3**	I		**STAT_STU_R: Power Status Bit[3]** ps bits of SHDSL ports 3 for STU-R
	PS_3	O	PP	**STAT_STU_C: Power Status Bit[3]** ps bits of SHDSL ports 3 for STU-C Buffer: C
	x_RXR_1_2	I		**xTALK: Rx Reference Signal[2], Channel 1**
	-	NU		**Normal: Not used**
P14	**PS_2**	I		**STAT_STU_R: Power Status Bit[2]** ps bits of SHDSL ports 2 for STU-R
	PS_2	O	PP	**STAT_STU_C: Power Status Bit[2]** ps bits of SHDSL ports 2 for STU-C Buffer: C
	x_RXR_1_1	I		**xTALK: Rx Reference Signal[1], Channel 1**
	-	NU		**Normal: Not used**
P15	**PS_1**	I		**STAT_STU_R: Power Status Bit[1]** ps bits of SHDSL ports 1 for STU-R
	PS_1	O	PP	**STAT_STU_C: Power Status Bit[1]** ps bits of SHDSL ports 1 for STU-C Buffer: C
	x_RXR_1_0	I		**xTALK: Rx Reference Signal[0], Channel 1**
	-	NU		**Normal: Not used**
P16	**PS_0**	I		**STAT_STU_R: Power Status Bit[0]** ps bits of SHDSL ports 0 for STU-R
	PS_0	O	PP	**STAT_STU_C: Power Status Bit[0]** ps bits of SHDSL ports 0 for STU-C Buffer: C
	x_REF	I		**xTALK: Clock Trigger Reference**
	-	NU		**Normal: Not used**
M13	**STATLINK**	O	PP	**STAT_STU_R: Status of Data Link** Buffer: C
	-	O	PP	**Not Used**
	x_RXR_2_6	I		**xTALK: Rx Reference Signal[6], Channel 2**
	-	NU		**Normal: Not used**
H13	**STAT_3**	O	PP	**STAT_STU_R: Status of SHDSL Port 3** Buffer: B
	STAT_3	O	PP	**STAT_STU_C: Status of SHDSL Port 3**
	x_TXR_3	O	PP	**xTALK: Transmit Reference Signal, Channel 3**
	-	I	PU1	**Normal: Not used**

Table 5 I/O Signals, SHDSL Status (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
J13	**STAT_2**	O	PP	**STAT_STU_R: Status of SHDSL Port 2** Buffer: B
	STAT_2	O	PP	**STAT_STU_C: Status of SHDSL Port 2**
	x_TXR_2	O	PP	**xTALK: Transmit Reference Signal, Channel 2**
	-	I	PU1	**Normal: Not used**
K13	**STAT_1**	O	PP	**STAT_STU_R: Status of SHDSL Port 1** Buffer: B
	STAT_1	O	PP	**STAT_STU_C: Status of SHDSL Port 1**
	x_TXR_1	O	PP	**xTALK: Transmit Reference Signal, Channel 1**
	-	I	PU1	**Normal: Not used**
L13	**STAT_0**	O	PP	**STAT_STU_R: Status of SHDSL Port 0** Buffer: B
	STAT_0	O	PP	**STAT_STU_C: Status of SHDSL Port 0**
	x_TXR_0	O	PP	**xTALK: Transmit Reference Signal, Channel 0**
	-	I	PU1	**Normal: Not used**

2.2.4 Host Interface

Not used inputs without an internal pull up resistor have to be connected to '1'.

Table 6 I/O Signals, Host Interface

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
R13	MPAD4	I		**Intel Demux: Address Bus[4]**
	MPAD4	I		**Motorola Async: Address Bus[4]**
	MPAD4	I		**Motorola Sync: Address Bus[4]**
	-	I	PU1	**RS232: Not Used**
	-	I	PU1	**SCI: Not Used**
R14	MPAD3	I		**Intel Demux: Address Bus[3]**
	MPAD3	I		**Motorola Async: Address Bus[3]**
	MPAD3	I		**Motorola Sync: Address Bus[3]**
	-	I	PU1	**RS232: Not Used**
	-	I	PU1	**SCI: Not Used**
R15	MPAD2	I		**Intel Demux: Address Bus[2]**
	MPAD2	I		**Motorola Async: Address Bus[2]**
	MPAD2	I		**Motorola Sync: Address Bus[2]**
	SPI_CI	I		**RS232: SPI_MODE_CI** SPI Clock Invert. Inverts SPI Clock Polarity 0_B **SPI_CLK_LOW**, Inactive State of SPICLK is low 1_B **SPI_CLK_HIGH**, Inactive State of SPICLK is high
	-	I	PU1	**SCI: Not Used**

Table 6 I/O Signals, Host Interface (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
R16	MPAD1	I		**Intel Demux: Address Bus[1]**
	MPAD1	I		**Motorola Async: Address Bus[1]**
	MPAD1	I		**Motorola Sync: Address Bus[1]**
	WDEN	I		**RS232: Watch-Dog Enable** 0_B **WD_DIS**, Watch-Dog disabled 1_B **WD_EN**, Watch-Dog enabled
	-	I	PU1	**SCI: Not Used**
R17	MPAD0	I		**Intel Demux: Address Bus[0]**
	MPAD0	I		**Motorola Async: Address Bus[0]**
	MPAD0	I		**Motorola Sync: Address Bus[0]**
	SPI_REV	I		**RS232: SPI_MODE_REV** Reverse Data. Determines the receive and transmit character bit order 0_B **SPI_REVERSE**, LSB of character sent and received first 1_B **SPI_NORMAL**, MSB of character sent and received first
	-	I	PU1	**SCI: Not Used**
A11	MPD7	I/O	TS	**Intel Demux: Data Bus Bit 7** Buffer: B
	MPD7	I/O	TS	**Motorola Async: Data Bus Bit 7**
	MPD7	I/O	TS	**Motorola Sync: Data Bus Bit 7**
	TXD	O	PP	**RS232: Serial Data Out**
	-	I	PU1	**SCI: Not Used**
B11	MPD6	I/O	TS	**Intel Demux: Data Bus Bit 6** Buffer: B
	MPD6	I/O	TS	**Motorola Async: Data Bus Bit 6**
	MPD6	I/O	TS	**Motorola Sync: Data Bus Bit 6**
	-	I	PU1	**RS232: Not Used**
	-	I	PU1	**SCI: Not Used**
C11	MPD5	I/O	TS	**Intel Demux: Data Bus Bit 5** Buffer: B
	MPD5	I/O	TS	**Motorola Async: Data Bus Bit 5**
	MPD5	I/O	TS	**Motorola Sync: Data Bus Bit 5**
	-	I	PU1	**RS232: Not Used**
	-	I	PU1	**SCI: Not Used**

Table 6 I/O Signals, Host Interface (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
D11	MPD4	I/O	TS	**Intel Demux: Data Bus Bit 4** Buffer: B
	MPD4	I/O	TS	**Motorola Async: Data Bus Bit 4**
	MPD4	I/O	TS	**Motorola Sync: Data Bus Bit 4**
	RXD	I		**RS232: Serial Data In**
	-	I	PU1	**SCI: Not Used**
D13	MPD3	I/O	TS	**Intel Demux: Data Bus Bit 3** Buffer: B
	MPD3	I/O	TS	**Motorola Async: Data Bus Bit 3**
	MPD3	I/O	TS	**Motorola Sync: Data Bus Bit 3**
	-	I	PU1	**RS232: Not used**
	-	I	PU1	**SCI: Not Used**
C13	MPD2	I/O	TS	**Intel Demux: Data Bus Bit 2** Buffer: B
	MPD2	I/O	TS	**Motorola Async: Data Bus Bit 2**
	MPD2	I/O	TS	**Motorola Sync: Data Bus Bit 2**
	-	I	PU1	**RS232: Not used**
	-	I	PU1	**SCI: Not Used**
D14	MPD1	I/O	TS	**Intel Demux: Data Bus Bit 1** Buffer: B
	MPD1	I/O	TS	**Motorola Async: Data Bus Bit 1**
	MPD1	I/O	TS	**Motorola Sync: Data Bus Bit 1**
	-	I	PU1	**RS232: Not used**
	-	I	PU1	**SCI: Not Used**
D15	MPD0	I/O	TS	**Intel Demux: Data Bus Bit 0** Buffer: B
	MPD0	I/O	TS	**Motorola Async: Data Bus Bit 0**
	MPD0	I/O	TS	**Motorola Sync: Data Bus Bit 0**
	-	I	PU1	**RS232: Not used**
	-	I	PU1	**SCI: Not Used**
R11	$\overline{\text{MPCS}}$	I		**Intel Demux: Chip Select**
	$\overline{\text{MPCS}}$	I		**Motorola Async: Chip Select**
	$\overline{\text{MPCS}}$	I		**Motorola Sync: Chip Select**
	$\overline{\text{SPISEL}}$	O	PP	**RS232: SPI Chip Select** Buffer: C
	DADDR_3	I		**SCI: Device Address Bit 3** To be latched in during hardware reset

Table 6 I/O Signals, Host Interface (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
T13	-	I		**Intel Demux: Not Used**
	-	I		**Motorola Async: Not Used**
	MPCLK	I		**Motorola Sync: Interface Clock**
	$\overline{WP}$	O	PP	**RS232: Write Protect** Buffer: C
	DADDR_2	I		**SCI: Device Address Bit 2** To be latched in during hardware reset
R12	$\overline{MPRDY}$	O	OD	**Intel Demux: Ready Signal** Buffer: B
	$\overline{MPDTACK}$	O	OD	**Motorola Async: Data Acknowledge**
	$\overline{MPTA}$	O	OD	**Motorola Sync: Transfer Acknowledge**
	SPICLK	O	PP	**RS232: SPI Serial Clock** up to 20 MHz
	-	NU		**SCI: Not Used**
T11	$\overline{MPWR}$	I		**Intel Demux: Write Signal**
	MPR/$\overline{W}$	I		**Motorola Async: Read/Write Signal**
	MPR	I		**Motorola Sync: Read/Write Acknowledge**
	SPIMISO	I		**RS232: SPI Master In, Slave Out (from FLASH)**
	DADDR_1	I		**SCI: Device Address Bit 1** To be latched in during hardware reset
U11	$\overline{MPRD}$	I		**Intel Demux: Read Signal**
	$\overline{MPDS}$	I		**Motorola Async: Data Strobe Signal**
	$\overline{MPTS}$	I		**Motorola Sync: Transfer Start Signal**
	-	NU		**RS232: Not Used**
	DADDR_0	I		**SCI: Device Address Bit 0**
V11	MPINT	O	TS	**Intel Demux: Interrupt** Open Source/Open Drain depending on active high/low Reset: active high Buffer: B
	MPINT	O	TS	**Motorola Async: Interrupt**
	MPINT	O	TS	**Motorola Sync: Interrupt**
	SPIMOSI	O	PP	**RS232: SPI Master Out, Slave In (to FLASH)**
	-	NU		**SCI: Not Used**

The different output characteristics of the pin SCDO can be set using a messages, after reset the output has a push pull characteristics.

Table 7 I/O Signals, Serial Control Interface (SCI)[1)]

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
R18	**SCCK**	I/O	PP	**NonBus: Clock of SCI** Buffer: C *Note: output for test purposes only*
	SCCK	I/O	PP	**Bus: Clock of SCI** *Note: output for test purposes only*
P18	**SCDO**	O	PP	**NonBus: Transmit Data of SCI** Buffer: C
	SCDO	O	OD	**Bus: Transmit Data of SCI**
T18	**SCDI**	I	-	**NonBus: Receive Data of SCI** Connect to '1' if input is not used
	SCDI	I	-	**Bus: Receive Data of SCI** Connect to '1' if input is not used

1) For debugging purposes it is recommended to connect these pins

2.2.5 System Interface 1

System Interface 1 can be operated in different modi, UTOPIA/POS-PHY mode, PHY mode and MAC mode, refer to **Chapter 1.3.3** for details. The ball assignment is specific to each of the modes and shown in detail in this section.

2.2.5.1 System Interface 1 in UTOPIA/POS-PHY Mode

The ball diagram for this application is shown in **Figure 14**. **Table 8** shows the System Interface 1 mode setting pins for operating this interface in UTOPIA/POS-PHY mode.

Table 8 I/O Signals, System Interface 1 Mode Settings UTOPIA/POS-PHY Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
K10	TCCONF2	I		**TC Configuration[2:0]** (Can be overwritten by Host) 0--_B **Do not use**, 100_B **UTOPIA L2 8-Bit**, 101_B **UTOPIA-2 L2 16-Bit**, 110_B **POSPHY-2**, 111_B **Do not use**,
J10	TCCONF1			
H10	TCCONF0			

Note: In UTOPIA L2 8-bit mode some of the pins are used for xTALK functionality. If this functionality is not enabled and not used in this mode all unused inputs have to be connected to '1'.

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B17	TXCLK	I		**UTOPIA L2 16-Bit: Transmit Clock**
	TXCLK	I		**UTOPIA L2 8-Bit: Transmit Clock**
	TFCLK	I		**POS-PHY L2: Transmit Clock**
A18	RXCLK	I		**UTOPIA L2 16-Bit: Receive Clock**
	RXCLK	I		**UTOPIA L2 8-Bit: Receive Clock**
	RFCLK	I		**POS-PHY L2: Receive Clock**
B16	TXPRTY	I		**UTOPIA L2 16-Bit: Transmit Parity**
	TXPRTY	I		**UTOPIA L2 8-Bit: Transmit Parity**
	TPRTY	I		**POS-PHY L2: Transmit Parity**
B15	-	I	PU1	**UTOPIA L2 16-Bit: Not used**
	-	I/O	PU1	**UTOPIA L2 8-Bit: Not used**
	TMOD	I		**POS-PHY L2: Transmit Word Modulo**
T15	-	I	PU1	**UTOPIA L2 16-Bit: Not used**
	-	I	PU1	**UTOPIA L2 8-Bit: Not used**
	TEOP	I		**POS-PHY L2: Transmit End of Packet**
A15	-	I	PU1	**UTOPIA L2 16-Bit: Not used**
	-	I	PU1	**UTOPIA L2 8-Bit: Not used**
	TERR	I		**POS-PHY L2: Transmit Error Indicator**
D18	TXSOC	I		**UTOPIA L2 16-Bit: Transmit Start of Cell**
	TXSOC	I		**UTOPIA L2 8-Bit: Transmit Start of Cell**
	TSOP	I		**POS-PHY L2: Transmit Start of Packet**
E18	$\overline{\text{TXENB}}$	I		**UTOPIA L2 16-Bit: Transmit Enable**
	$\overline{\text{TXENB}}$	I		**UTOPIA L2 8-Bit: Transmit Enable**
	$\overline{\text{TENB}}$	I		**POS-PHY L2: Transmit Enable**
H16	$\overline{\text{RXENB}}$	I		**UTOPIA L2 16-Bit: Receive Enable**
	$\overline{\text{RXENB}}$	I		**UTOPIA L2 8-Bit: Receive Enable**
	$\overline{\text{RENB}}$	I		**POS-PHY L2: Receive Enable**
G16	TXADDR4	I		**UTOPIA L2 16-Bit: Transmit PHY Address[4]**
	TXADDR4	I		**UTOPIA L2 8-Bit: Transmit PHY Address[4]**
	TADDR4	I		**POS-PHY L2: Transmit PHY Address[4]**
F16	TXADDR3	I		**UTOPIA L2 16-Bit: Transmit PHY Address[3]**
	TXADDR3	I		**UTOPIA L2 8-Bit: Transmit PHY Address[3]**
	TADDR3	I		**POS-PHY L2: Transmit PHY Address[3]**
E16	TXADDR2	I		**UTOPIA L2 16-Bit: Transmit PHY Address[2]**
	TXADDR2	I		**UTOPIA L2 8-Bit: Transmit PHY Address[2]**
	TADDR2	I		**POS-PHY L2: Transmit PHY Address[2]**

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
D16	TXADDR1	I		**UTOPIA L2 16-Bit: Transmit PHY Address[1]**
	TXADDR1	I		**UTOPIA L2 8-Bit: Transmit PHY Address[1]**
	TADDR1	I		**POS-PHY L2: Transmit PHY Address[1]**
C17	TXADDR0	I		**UTOPIA L2 16-Bit: Transmit PHY Address[0]**
	TXADDR0	I		**UTOPIA L2 8-Bit: Transmit PHY Address[0]**
	TADDR0	I		**POS-PHY L2: Transmit PHY Address[0]**
V12	RXADDR4	I		**UTOPIA L2 16-Bit: Receive PHY Address[4]**
	RXADDR4	I		**UTOPIA L2 8-Bit: Receive PHY Address[4]**
	RADDR4	I		**POS-PHY L2: Receive PHY Address[4]**
T12	RXADDR3	I		**UTOPIA L2 16-Bit: Receive PHY Address[3]**
	RXADDR3	I		**UTOPIA L2 8-Bit: Receive PHY Address[3]**
	RADDR3	I		**POS-PHY L2: Receive PHY Address[3]**
U13	RXADDR2	I		**UTOPIA L2 16-Bit: Receive PHY Address[2]**
	RXADDR2	I		**UTOPIA L2 8-Bit: Receive PHY Address[2]**
	RADDR2	I		**POS-PHY L2: Receive PHY Address[2]**
C12	RXADDR1	I		**UTOPIA L2 16-Bit: Receive PHY Address[1]**
	RXADDR1	I		**UTOPIA L2 8-Bit: Receive PHY Address[1]**
	RADDR1	I		**POS-PHY L2: Receive PHY Address[1]**
D12	RXADDR0	I		**UTOPIA L2 16-Bit: Receive PHY Address[0]**
	RXADDR0	I		**UTOPIA L2 8-Bit: PHY Address[0]**
	RADDR0	I		**POS-PHY L2: Receive PHY Address[0]**
A17	TXDATA15	I		**UTOPIA L2 16-Bit: Transmit Data[15]**
	x_RXR_2_5	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[5], Channel 2**
	TDAT15	I		**POS-PHY L2: Transmit Data[15]**
E15	TXDATA14	I		**UTOPIA L2 16-Bit: Transmit Data[14]**
	x_RXR_2_4	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[4], Channel 2**
	TDAT14	I		**POS-PHY L2: Transmit Data[14]**
E14	TXDATA13	I		**UTOPIA L2 16-Bit: Transmit Data[13]**
	x_RXR_2_3	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[3], Channel 2**
	TDAT13	I		**POS-PHY L2: Transmit Data[13]**
E13	TXDATA12	I		**UTOPIA L2 16-Bit: Transmit Data[12]**
	x_RXR_2_2	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[2], Channel 2**
	TDAT12	I		**POS-PHY L2: Transmit Data[12]**

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
E12	TXDATA11	I		**UTOPIA L2 16-Bit: Transmit Data[11]**
	x_RXR_2_1	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[1], Channel 2**
	TDAT11	I		**POS-PHY L2: Transmit Data[11]**
E11	TXDATA10	I		**UTOPIA L2 16-Bit: Transmit Data[10]**
	x_RXR_2_0	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[0], Channel 2**
	TDAT10	I		**POS-PHY L2: Transmit Data[10]**
F13	TXDATA9	I		**UTOPIA L2 16-Bit: Transmit Dat[9]**
	x_RXR_3_6	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[6], Channel 3**
	TDAT9	I		**POS-PHY L2: Transmit Data[9]**
G13	TXDATA8	I	PU1	**UTOPIA L2 16-Bit: Transmit Data[8]**
	x_RXR_3_5	I	PU1	**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[5], Channel 3**
	TDAT8	I	PU1	**POS-PHY L2: Transmit Data[8]**
N18	TXDATA7	I		**UTOPIA L2 16-Bit: Transmit Data[7]**
	TXDATA7	I		**UTOPIA L2 8-Bit: Transmit Data[7]**
	TDAT7	I		**POS-PHY L2: Transmit Data[7]**
M18	TXDATA6	I		**UTOPIA L2 16-Bit: Transmit Data[6]**
	TXDATA6	I		**UTOPIA L2 8-Bit: Transmit Data[6]**
	TDAT6	I		**POS-PHY L2: Transmit Data[6]**
L18	TXDATA5	I		**UTOPIA L2 16-Bit: Transmit Data[5]**
	TXDATA5	I		**UTOPIA L2 8-Bit: Transmit Data[5]**
	TDAT5	I		**POS-PHY L2: Transmit Data[5]**
K18	TXDATA4	I		**UTOPIA L2 16-Bit: Transmit Data[4]**
	TXDATA4	I		**UTOPIA L2 8-Bit: Transmit Data[4]**
	TDAT4	I		**POS-PHY L2: Transmit Data[4]**
J18	TXDATA3	I		**UTOPIA L2 16-Bit: Transmit Data[3]**
	TXDATA3	I		**UTOPIA L2 8-Bit: Transmit Data[3]**
	TDAT3	I		**POS-PHY L2: Transmit Data[3]**
H18	TXDATA2	I		**UTOPIA L2 16-Bit: Transmit Data[2]**
	TXDATA2	I		**UTOPIA L2 8-Bit: Transmit Data[2]**
	TDAT2	I		**POS-PHY L2: Transmit Data[2]**
G18	TXDATA1	I		**UTOPIA L2 16-Bit: Transmit Data[1]**
	TXDATA1	I		**UTOPIA L2 8-Bit: Transmit Data[1]**
	TDAT1	I		**POS-PHY L2: Transmit Data[1]**

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F18	TXDATA0	I		**UTOPIA L2 16-Bit: Transmit Data[0]**
	TXDATA0	I		**UTOPIA L2 8-Bit: Transmit Data[0]**
	TDAT0	I		**POS-PHY L2: Transmit Data[0]**
B18	$\overline{\text{TXFULL}}$/ TXCLAV	O	TS	**UTOPIA L2 16-Bit: Transmit FIFO full/Cell Buffer available**
	$\overline{\text{TXFULL}}$/ TXCLAV	O	TS	**UTOPIA L2 8-Bit: Transmit FIFO full/Cell Buffer available**
	PTPA	O	TS	**POS-PHY L2: Polled PHY Transmit Packet Available**
C18	$\overline{\text{RXEMPTY}}$/ RXCLAV	O	TS	**UTOPIA L2 16-Bit: Receive FIFO empty/Cell Buffer available**
	$\overline{\text{RXEMPTY}}$/ RXCLAV	O	TS	**UTOPIA L2 8-Bit: Receive FIFO empty/Cell Buffer available**
	PRPA	O	TS	**POS-PHY L2: Polled PHY Receive Packet Available**
J16	RXSOC	O	TS	**UTOPIA L2 16-Bit: Receive Start of Cell**
	RXSOC	O	TS	**UTOPIA L2 8-Bit: Receive Start of Cell**
	RSOP	O	TS	**POS-PHY L2: Receive Start of Packet**
B13	RXPRTY	O	TS	**UTOPIA L2 16-Bit: Receive Parity**
	RXPRTY	O	TS	**UTOPIA L2 8-Bit: Receive Parity**
	RPRTY	O	TS	**POS-PHY L2: Receive Parity**
K16	RXDATA15	O	TS	**UTOPIA L2 16-Bit: Receive Data[15]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT15	O	TS	**POS-PHY L2: Receive Data[15]**
L16	RXDATA14	O	TS	**UTOPIA L2 16-Bit: Receive Data[14]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT14	O	TS	**POS-PHY L2: Receive Data[14]**
M16	RXDATA13	O	TS	**UTOPIA L2 16-Bit: Receive Data[13]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT13	O	TS	**POS-PHY L2: Receive Data[13]**
N16	RXDATA12	O	TS	**UTOPIA L2 16-Bit: Receive Data[12]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT12	O	TS	**POS-PHY L2: Receive Data[12]**
P17	RXDATA11	O	TS	**UTOPIA L2 16-Bit: Receive Data[11]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT11	O	TS	**POS-PHY L2: Receive Data[11]**
E17	RXDATA10	O	TS	**UTOPIA L2 16-Bit: Receive Data[10]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT10	O	TS	**POS-PHY L2: Receive Data[10]**

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
T17	RXDATA9	O	TS	**UTOPIA L2 16-Bit: Receive Data[9]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT9	O	TS	**POS-PHY L2: Receive Data[9]**
A16	RXDATA8	O	TS	**UTOPIA L2 16-Bit: Receive Data[8]**
	-	O	PP	**UTOPIA L2 8-Bit: Not used**
	RDAT8	O	TS	**POS-PHY L2: Receive Data[8]**
N17	RXDATA7	O	TS	**UTOPIA L2 16-Bit: Receive Data[7]**
	RXDATA7	O	TS	**UTOPIA L2 8-Bit: Receive Data[7]**
	RDAT7	O	TS	**POS-PHY L2: Receive Data[7]**
M17	RXDATA6	O	TS	**UTOPIA L2 16-Bit: Receive Data[6]**
	RXDATA6	O	TS	**UTOPIA L2 8-Bit: Receive Data[6]**
	RDAT6	O	TS	**POS-PHY L2: Receive Data[6]**
L17	RXDATA5	O	TS	**UTOPIA L2 16-Bit: Receive Data[5]**
	RXDATA5	O	TS	**UTOPIA L2 8-Bit: Receive Data[5]**
	RDAT5	O	TS	**POS-PHY L2: Receive Data[5]**
K17	RXDATA4	O	TS	**UTOPIA L2 16-Bit: Receive Data[4]**
	RXDATA4	O	TS	**UTOPIA L2 8-Bit: Receive Data[4]**
	RDAT4	O	TS	**POS-PHY L2: Receive Data[4]**
J17	RXDATA3	O	TS	**UTOPIA L2 16-Bit: Receive Data[3]**
	RXDATA3	O	TS	**UTOPIA L2 8-Bit: Receive Data[3]**
	RDAT3	O	TS	**POS-PHY L2: Receive Data[3]**
H17	RXDATA2	O	TS	**UTOPIA L2 16-Bit: Receive Data[2]**
	RXDATA2	O	TS	**UTOPIA L2 8-Bit: Receive Data[2]**
	RDAT2	O	TS	**POS-PHY L2: Receive Data[2]**
G17	RXDATA1	O	TS	**UTOPIA L2 16-Bit: Receive Data[1]**
	RXDATA1	O	TS	**UTOPIA L2 8-Bit: Receive Data[1]**
	RDAT1	O	TS	**POS-PHY L2: Receive Data[1]**
F17	RXDATA0	O	TS	**UTOPIA L2 16-Bit: Receive Data[0]**
	RXDATA0	O	TS	**UTOPIA L2 8-Bit: Receive Data[0]**
	RDAT0	O	TS	**POS-PHY L2: Receive Data[0]**
C16	-	O	PP	**UTOPIA L2 16-Bit: Not used**
	x_RXR_3_4	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[4], Channel 3**
	STPA	O	TS	**POS-PHY L2: Selected PHY Transmit Packet Available**
C14	-	O	PP	**UTOPIA L2 16-Bit: Not used**
	x_RXR_3_3	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[3], Channel 3**
	RVAL	O	TS	**POS-PHY L2: Receive Data Valid**

Table 9 I/O Signals, System Interface 1 (Parallel Data Interface), UTOPIA/POS-PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
T14	-	O	PP	**UTOPIA L2 16-Bit: Not used**
	x_RXR_3_2	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[2], Channel 3**
	RMOD	O	TS	**POS-PHY L2: Receive Word Modulo**
V15	-	O	PP	**UTOPIA L2 16-Bit: Not used**
	x_RXR_3_1	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[1], Channel 3**
	REOP	O	TS	**POS-PHY L2: Receive End Of Packet**
V14	-	O	PP	**UTOPIA L2 16-Bit: Not used**
	x_RXR_3_0	I		**UTOPIA L2 8-Bit: xTALK; Rx Reference Signal[0], Channel 3**
	RERR	O	TS	**POS-PHY L2: Receive Error Indicator**

2.2.5.2 System Interface 1 in PHY Mode

The ball diagram of this application is shown in **Figure 15**. **Table 10** shows the System Interface 1 mode setting pins for operating the interface in PHY mode, **Table 12** contains not used balls in this mode. For this mode the pin PHY (ball T15) has to be set to '1' and the pin MDIOSLAVE (ball D16) to '1' as well.

Table 10 I/O Signals, System Interface 1 Mode Settings PHY Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
K10	TCCONF2	I		**TC Configuration[2:0]** (Can be overwritten by Host) 000_B **MII**, 001_B **RMII**, 011_B **SMII source synchronous (SS-SMII)**, 010_B **Do not use**, $1--_B$ **Do not use**,
J10	TCCONF1			
H10	TCCONF0			

Note: In PHY mode some of the pins are used for xTALK functionality. If this functionality is not enabled and not used in this mode all unused inputs have to be connected to '1'.

In the following table the term SMII stands for SMII source synchronous mode.

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B17	-	I	PU1	**MII PHY: Not used**
	TX_EN_3	I		**RMII PHY: Transmit Enable Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	TX_CLK	I		**SMII PHY: Transmit Clock** 125 MHz only used in source synchronous mode

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
A18	-	I	PU1	**MII PHY: Not used**
	TX_EN_2	I		**RMII PHY: Transmit Enable Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII PHY: Not used**
T15	PHY	I		**MII PHY: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
	PHY	I		**RMII PHY: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
	PHY	I		**SMII PHY: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
D18	-	I	PU1	**MII PHY: Not used**
	TX_EN_1	I		**RMII PHY: Transmit Enable Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII PHY: Not used**
E18	TX_EN	I		**MII PHY: Transmit Enable**
	TX_EN_0	I		**RMII PHY: Transmit Enable Port 0** Connect to '1' if channel is not used
	TX_SYNC	I		**SMII PHY: Transmit Sync**
E16	**TX_ER**	I		**MII PHY: Transmit Coding Error**
	-	I	PU1	**RMII PHY: Not used**
	-	I	PU1	**SMII PHY: Not used**
D16	MDIOSLAVE	I		**MII PHY: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode 1_B **SLAVE**, MDIO in Slave (PHY) Mode
	MDIOSLAVE	I		**RMII PHY: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode, MDC output 1_B **SLAVE**, MDIO in Slave (PHY) Mode, MDC input
	MDIOSLAVE	I		**SMII PHY: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode, MDC output 1_B **SLAVE**, MDIO in Slave (PHY) Mode, MDC input
C17	-	I	PU1	**MII PHY: Not used**
	REFCLK_i	I		**RMII PHY: Reference Clock in** 50 MHz
	CLOCK_i	I		**SMII PHY: Reference Clock in** 125 MHz

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
V12	-	I	PU1	**MII PHY: Not used**
	-	I	PU1	**RMII PHY: Not used**
	SYNC_i	I		**SMII PHY: SYNC input signal** Used for cascading over chip boundaries
U13	PHYADDR4	I	PU1	**MII PHY: MDIO PHY Address[4]**
	PHYADDR4	I	PU1	**RMII PHY: MDIO PHY Address[4]**
	PHYADDR4	I	PU1	**SMII PHY: MDIO PHY Address[4]**
C12	PHYADDR3	I	PU1	**MII PHY: MDIO PHY Address[3]**
	PHYADDR3	I	PU1	**RMII PHY: MDIO PHY Address[3]**
	PHYADDR3	I	PU1	**SMII PHY: MDIO PHY Address[3]**
D12	PHYADDR2	I	PU1	**MII PHY: MDIO PHY Address[2]**
	PHYADDR2	I	PU1	**RMII PHY: MDIO PHY Address[2]**
	PHYADDR2	I	PU1	**SMII PHY: MDIO PHY Address[2]**
A17	x_RXR_2_5	I		**MII PHY: xTALK; Rx Reference Signal[5], Channel 2**
	x_RXR_2_5	I		**RMII PHY: xTALK; Rx Reference Signal[5], Channel 2**
	x_RXR_2_5	I		**SMII PHY: xTALK; Rx Reference Signal[5], Channel 2**
E15	x_RXR_2_4	I		**MII PHY: xTALK; Rx Reference Signal[4], Channel 2**
	x_RXR_2_4	I		**RMII PHY: xTALK; Rx Reference Signal[4], Channel 2**
	x_RXR_2_4	I		**SMII PHY: xTALK; Rx Reference Signal[4], Channel 2**
E14	x_RXR_2_3	I		**MII PHY: xTALK; Rx Reference Signal[3], Channel 2**
	x_RXR_2_3	I		**RMII PHY: xTALK; Rx Reference Signal[3], Channel 2**
	x_RXR_2_3	I		**SMII PHY: xTALK; Rx Reference Signal[3], Channel 2**
E13	x_RXR_2_2	I		**MII PHY: xTALK; Rx Reference Signal[2], Channel 2**
	x_RXR_2_2	I		**RMII PHY: xTALK; Rx Reference Signal[2], Channel 2**
	x_RXR_2_2	I		**SMII PHY: xTALK; Rx Reference Signal[2], Channel 2**
E12	x_RXR_2_1	I		**MII PHY: xTALK; Rx Reference Signal[1], Channel 2**
	x_RXR_2_1	I		**RMII PHY: xTALK; Rx Reference Signal[1], Channel 2**
	x_RXR_2_1	I		**SMII PHY: xTALK; Rx Reference Signal[1], Channel 2**
E11	x_RXR_2_0	I		**MII PHY: xTALK; Rx Reference Signal[0], Channel 2**
	x_RXR_2_0	I		**RMII PHY: xTALK; Rx Reference Signal[0], Channel 2**
	x_RXR_2_0	I		**SMII PHY: xTALK; Rx Reference Signal[0], Channel 2**
F13	x_RXR_3_6	I		**MII PHY: xTALK; Rx Reference Signal[6], Channel 3**
	x_RXR_3_6	I		**RMII PHY: xTALK; Rx Reference Signal[6], Channel 3**
	x_RXR_3_6	I		**SMII PHY: xTALK; Rx Reference Signal[6], Channel 3**
G13	x_RXR_3_5	I	PU1	**MII PHY: xTALK; Rx Reference Signal[5], Channel 3**
	x_RXR_3_5	I	PU1	**RMII PHY: xTALK; Rx Reference Signal[5], Channel 3**
	x_RXR_3_5	I	PU1	**SMII PHY: xTALK; Rx Reference Signal[5], Channel 3**

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
N18	-	I	PU1	**MII PHY: Not used**
	TXD3_1	I		**RMII PHY: Transmit Data[1] Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	-	I	PU1	**SMII PHY: Not used**
M18	-	I	PU1	**MII PHY: Not used**
	TXD3_0	I		**RMII PHY: Transmit Data[0] Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	TX_3	I		**SMII PHY: Transmit Data Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
L18	-	I	PU1	**MII PHY: Not used**
	TXD2_1	I		**RMII PHY: Transmit Data[1] Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII PHY: Not used**
K18	-	I	PU1	**MII PHY: Not used**
	TXD2_0	I		**RMII PHY: Transmit Data[0] Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	TX_2	I		**SMII PHY: Transmit Data Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
J18	TXD3	I		**MII PHY: Transmit Data[3]**
	TXD1_1	I		**RMII PHY: Transmit Data[1] Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII PHY: Not used**
H18	TXD2	I		**MII PHY: Transmit Data[2]**
	TXD1_0	I		**RMII PHY: Transmit Data[0] Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	TX_1	I		**SMII PHY: Transmit Data Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G18	TXD1	I		**MII PHY: Transmit Data[1]**
	TXD0_1	I		**RMII PHY: Transmit Data[1] Port 0** Connect to '1' if channel is not used
	-	I	PU1	**SMII PHY: Not used**

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F18	TXD0	I		**MII PHY: Transmit Data[0]**
	TXD0_0	I		**RMII PHY: Transmit Data[0] Port 0** Connect to '1' if channel is not used
	TX_0	I		**SMII PHY: Transmit Data Port 0** Connect to '1' if channel is not used
B18	**TX_CLK**	O	PP	**MII PHY: Transmit Clock** 2.5 MHz for 10 Mbit/s xMII 25 MHz for 100 Mbit/s xMII
	-	O	PP	**RMII PHY: Not used**
	RX_CLK	O	PP	**SMII PHY: Receive Clock** 125 MHz
C18	RX_CLK	O	PP	**MII PHY: Receive Clock** 2.5 MHz for 10 Mbit/s xMII 25 MHz for 100 Mbit/s xMII +/- 100 ppm duty cycle: 35-65%
	REFCLK_o	O	PP	**RMII PHY: Reference Clock** Derived from internal crystal: 50 MHz
	CLOCK_o	O	PP	**SMII PHY: Reference Clock** Derived from internal crystal: 125 MHz
J16	-	O	PP	**MII PHY: Not used**
	RX_ER_3	O	PP	**RMII PHY: Receive Error Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	SYNC_o	O	PP	**SMII PHY: SYNC output signal** Used for cascading over chip boundaries
B13	-	O	PP	**MII PHY: Not used**
	RX_ER_2	O	PP	**RMII PHY: Receive Error Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
K16	-	O	PP	**MII PHY: Not used**
	RX_ER_1	O	PP	**RMII PHY: Receive Error Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
L16	RX_ER	O	PP	**MII PHY: Receive Error**
	RX_ER_0	O	PP	**RMII PHY: Receive Error Port 0** Do not connect if channel is not used
	-	O	PP	**SMII PHY: Not used**

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
M16	RX_DV	O	PP	**MII PHY: Receive Data Valid**
	CRS_DV_3	O	PP	**RMII PHY: Carrier Sense/Receive Data Valid Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
N16	COL	O	PP	**MII PHY: Collision Detected**
	CRS_DV_2	O	PP	**RMII PHY: Carrier Sense/Receive Data Valid Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
P17	CRS	O	PP	**MII PHY: Carrier Sense**
	CRS_DV_1	O	PP	**RMII PHY: Carrier Sense/Receive Data Valid Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
E17	-	O	PP	**MII PHY: Not used**
	CRS_DV_0	O	PP	**RMII PHY: Carrier Sense/Receive Data Valid Port 0** Do not connect if channel is not used
	RX_SYNC	O	PP	**SMII PHY: Receive Sync**
T17	MDC	I/O	PP	**MII PHY: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
	MDC	I/O	PP	**RMII PHY: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
	MDC	I/O	PP	**SMII PHY: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
A16	MDIO	I/O	TS	**MII PHY: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
	MDIO	I/O	TS	**RMII PHY: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
	MDIO	I/O	TS	**SMII PHY: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
N17	-	O	PP	**MII PHY: Not used**
	RXD3_1	O	PP	**RMII PHY: Receive Data[1] Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	-	O	PP	**SMII PHY: Not used**

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
M17	-	O	PP	**MII PHY: Not used**
	RXD3_0	O	PP	**RMII PHY: Receive Data[0] Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	RX_3	O	PP	**SMII PHY: Receive Data Port 3** Do not connect for SOCRATES™-1e or if channel is not used
L17	-	O	PP	**MII PHY: Not used**
	RXD2_1	O	PP	**RMII PHY: Receive Data[1] Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
K17	-	O	PP	**MII PHY: Not used**
	RXD2_0	O	PP	**RMII PHY: Receive Data[0] Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	RX_2	O	PP	**SMII PHY: Receive Data Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
J17	RXD3	O	PP	**MII PHY: Receive Data[3]**
	RXD1_1	O	PP	**RMII PHY: Receive Data[1] Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII PHY: Not used**
H17	RXD2	O	PP	**MII PHY: Receive Data[2]**
	RXD1_0	O	PP	**RMII PHY: Receive Data[0] Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	RX_1	O	PP	**SMII PHY: Receive Data Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G17	RXD1	O	PP	**MII PHY: Receive Data[1]**
	RXD0_1	O	PP	**RMII PHY: Receive Data[1] Port 0** Do not connect if channel is not used
	-	O	PP	**SMII PHY: Not used**
F17	RXD0	O	PP	**MII PHY: Receive Data[0]**
	RXD0_0	O	PP	**RMII PHY: Receive Data[0] Port 0** Do not connect if channel is not used
	RX_0	O	PP	**SMII PHY: Receive Data Port 0** Do not connect if channel is not used
C16	x_RXR_3_4	I		**MII PHY: xTALK; Rx Reference Signal[4], Channel 3**
	x_RXR_3_4	I		**RMII PHY: xTALK; Rx Reference Signal[4], Channel 3**
	x_RXR_3_4	I		**SMII PHY: xTALK; Rx Reference Signal[4], Channel 3**

Table 11 I/O Signals, System Interface 1 (Parallel Data Interface), PHY Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C14	x_RXR_3_3	I		**MII PHY: xTALK; Rx Reference Signal[3], Channel 3**
	x_RXR_3_3	I		**RMII PHY: xTALK; Rx Reference Signal[3], Channel 3**
	x_RXR_3_3	I		**SMII PHY: xTALK; Rx Reference Signal[3], Channel 3**
T14	x_RXR_3_2	I		**MII PHY: xTALK; Rx Reference Signal[2], Channel 3**
	x_RXR_3_2	I		**RMII PHY: xTALK; Rx Reference Signal[2], Channel 3**
	x_RXR_3_2	I		**SMII PHY: xTALK; Rx Reference Signal[2], Channel 3**
V15	x_RXR_3_1	I		**MII PHY: xTALK; Rx Reference Signal[1], Channel 3**
	x_RXR_3_1	I		**RMII PHY: xTALK; Rx Reference Signal[1], Channel 3**
	x_RXR_3_1	I		**SMII PHY: xTALK; Rx Reference Signal[1], Channel 3**
V14	x_RXR_3_0	I		**MII PHY: xTALK; Rx Reference Signal[0], Channel 3**
	x_RXR_3_0	I		**RMII PHY: xTALK; Rx Reference Signal[0], Channel 3**
	x_RXR_3_0	I		**SMII PHY: xTALK; Rx Reference Signal[0], Channel 3**

Table 12 Not Used Pins, PHY Mode, Internal Pull Up

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B16, B15, A15, H16, G16, F16, T12	PU	I	PU1	**Not used**

2.2.5.3 System Interface 1 in MAC Mode

The ball diagram of this application is shown in **Figure 16**. **Table 13** shows the System Interface 1 mode setting pins for operating this interface in MAC mode, **Table 15** contains not used balls in this mode. For this mode the pin PHY (ball T15) has to be set to '0' and the pin MDIOSLAVE (ball D16) to '0' as well. Note that only MII and SS-MII as interfaces are supported in MAC Mode.

Table 13 I/O Signals, System Interface 1 Mode Settings MAC Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
K10	TCCONF2	I		**TC Configuration[2:0]** (Can be overwritten by Host) 000_B **MII**, 010_B **SMII typical (SMII)**, 011_B **SMII source synchronous (SS-SMII)**, $1--_B$ **Do not use**, 010_B **Do not use**,
J10	TCCONF1			
H10	TCCONF0			

Note: In MAC mode some of the pins are used for xTALK functionality. If this functionality is not enabled and not used in this mode all unused inputs have to be connected to '1'. Other 'Not Used' inputs without an internal pull up resistor also have to be connected to '1'.

In the following table the term SMII stands for SMII source synchronous mode. The RMII mode is not supported in MAC mode, hence this mode needs to be ignored in the following table.

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B17	RX_CLK	I		**MII MAC: Receive Clock** 2.5 MHz and 25 MHz
	CRS_DV_3	I		**RMII MAC: Carrier Sense/Receive Data Valid Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	RX_CLK	I		**SMII MAC: Receive Clock** 125 MHz
A18	TX_CLK	I		**MII MAC: Transmit Clock** 2.5 MHz and 25 MHz
	CRS_DV_2	I		**RMII MAC: Carrier Sense/Receive Data Valid Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
B16	**COL**	I		**MII MAC: Collision Detected**
	-	I	PU1	**RMII MAC: Not used**
	-	I	PU1	**SMII MAC: Not used**
T15	PHY	I		**MII MAC: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
	PHY	I		**RMII MAC: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
	PHY	I		**SMII MAC: PHY Mode of xMII data path** 0_B **MAC**, MAC Mode 1_B **PHY**, PHY Mode
D18	CRS	I		**MII MAC: Carrier Sense**
	CRS_DV_1	I		**RMII MAC: Carrier Sense/Receive Data Valid Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
E18	RX_DV	I		**MII MAC: Receive Data Valid**
	CRS_DV_0	I		**RMII MAC: Carrier Sense/Receive Data Valid Port 0** Connect to '1' if channel is not used
	RX_SYNC	I		**SMII MAC: Receive Sync**
H16	-	I	PU1	**MII MAC: Not used**
	RX_ER_3	I		**RMII MAC: Receive Error Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	-	I		**SMII MAC: Not used**

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
G16	-	I	PU1	**MII MAC: Not used**
	RX_ER_2	I		**RMII MAC: Receive Error Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
F16	-	I	PU1	**MII MAC: Not used**
	RX_ER_1	I		**RMII MAC: Receive Error Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
E16	RX_ER	I		**MII MAC: Receive Error**
	RX_ER_0	I		**RMII MAC: Receive Error Port 0** Connect to '1' if channel is not used
	-	I	PU1	**SMII MAC: Not used**
D16	MDIOSLAVE	I		**MII MAC: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode 1_B **SLAVE**, MDIO in Slave (PHY) Mode
	MDIOSLAVE	I		**RMII MAC: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode, MDC output 1_B **SLAVE**, MDIO in Slave (PHY) Mode, MDC input
	MDIOSLAVE	I		**SMII MAC: Slave Mode of MDIO** 0_B **MASTER**, MDIO in Master (MAC) Mode, MDC output 1_B **SLAVE**, MDIO in Slave (PHY) Mode, MDC input
C17	-	I	PU1	**MII MAC: Not used**
	REFCLK_i	I		**RMII MAC: Reference Clock in** 50 MHz
	CLOCK_i	I		**SMII MAC: Reference Clock in** 125 MHz
V12	-	I	PU1	**MII MAC: Not used**
	-	I	PU1	**RMII MAC: Not used**
	SYNC_i	I		**SMII MAC: SYNC input signal** Used for cascading over chip boundaries Connect to '1' if not used
U13	PHYADDR4	I	PU1	**MII MAC: Used in Virtual PHY Mode only**
	PHYADDR4	I	PU1	**RMII MAC: Used in Virtual PHY Mode only**
	PHYADDR4	I	PU1	**SMII MAC: Used in Virtual PHY Mode only**
C12	PHYADDR3	I	PU1	**MII MAC: Used in Virtual PHY Mode only**
	PHYADDR3	I	PU1	**RMII MAC: Used in Virtual PHY Mode only**
	PHYADDR3	I	PU1	**SMII MAC: Used in Virtual PHY Mode only**

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
D12	PHYADDR2	I	PU1	**MII MAC: Used in Virtual PHY Mode only**
	PHYADDR2	I	PU1	**RMII MAC: Used in Virtual PHY Mode only**
	PHYADDR2	I	PU1	**SMII MAC: Used in Virtual PHY Mode only**
A17	x_RXR_2_5	I		**MII MAC: xTALK; Rx Reference Signal[5], Channel 2**
	x_RXR_2_5	I		**RMII MAC: xTALK; Rx Reference Signal[5], Channel 2**
	x_RXR_2_5	I		**SMII MAC: xTALK; Rx Reference Signal[5], Channel 2**
E15	x_RXR_2_4	I		**MII MAC: xTALK; Rx Reference Signal[4], Channel 2**
	x_RXR_2_4	I		**RMII MAC: xTALK; Rx Reference Signal[4], Channel 2**
	x_RXR_2_4	I		**SMII MAC: xTALK; Rx Reference Signal[4], Channel 2**
E14	x_RXR_2_3	I		**MII MAC: xTALK; Rx Reference Signal[3], Channel 2**
	x_RXR_2_3	I		**RMII MAC: xTALK; Rx Reference Signal[3], Channel 2**
	x_RXR_2_3	I		**SMII MAC: xTALK; Rx Reference Signal[3], Channel 2**
E13	x_RXR_2_2	I		**MII MAC: xTALK; Rx Reference Signal[2], Channel 2**
	x_RXR_2_2	I		**RMII MAC: xTALK; Rx Reference Signal[2], Channel 2**
	x_RXR_2_2	I		**SMII MAC: xTALK; Rx Reference Signal[2], Channel 2**
E12	x_RXR_2_1	I		**MII MAC: xTALK; Rx Reference Signal[1], Channel 2**
	x_RXR_2_1	I		**RMII MAC: xTALK; Rx Reference Signal[1], Channel 2**
	x_RXR_2_1	I		**SMII MAC: xTALK; Rx Reference Signal[1], Channel 2**
E11	x_RXR_2_0	I		**MII MAC: xTALK; Rx Reference Signal[0], Channel 2**
	x_RXR_2_0	I		**RMII MAC: xTALK; Rx Reference Signal[0], Channel 2**
	x_RXR_2_0	I		**SMII MAC: xTALK; Rx Reference Signal[0], Channel 2**
F13	x_RXR_3_6	I		**MII MAC: xTALK; Rx Reference Signal[6], Channel 3**
	x_RXR_3_6	I		**RMII MAC: xTALK; Rx Reference Signal[6], Channel 3**
	x_RXR_3_6	I		**SMII MAC: xTALK; Rx Reference Signal[6], Channel 3**
G13	x_RXR_3_5	I	PU1	**MII MAC: xTALK; Rx Reference Signal[5], Channel 3**
	x_RXR_3_5	I	PU1	**RMII MAC: xTALK; Rx Reference Signal[5], Channel 3**
	x_RXR_3_5	I	PU1	**SMII MAC: xTALK; Rx Reference Signal[5], Channel 3**
N18	-	I	PU1	**MII MAC: Not used**
	RXD3_1	I		**RMII MAC: Receive Data[1] Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
M18	-	I	PU1	**MII MAC: Not used**
	RXD3_0	I		**RMII MAC: Receive Data[0] Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used
	RX_3	I		**SMII MAC: Receive Data Port 3** Connect to '1' for SOCRATES™-1e or if channel is not used

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
L18	-	I	PU1	**MII MAC: Not used**
	RXD2_1	I		**RMII MAC: Receive Data[1] Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
K18	-	I	PU1	**MII MAC: Not used**
	RXD2_0	I		**RMII MAC: Receive Data[0] Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	RX_2	I		**SMII MAC: Receive Data Port 2** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
J18	RXD3	I		**MII MAC: Receive Data[3]**
	RXD1_1	I		**RMII MAC: Receive Data[1] Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	I	PU1	**SMII MAC: Not used**
H18	RXD2	I		**MII MAC: Receive Data[2]**
	RXD1_0	I		**RMII MAC: Receive Data[0] Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	RX_1	I		**SMII MAC: Receive Data Port 1** Connect to '1' for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G18	RXD1	I		**MII MAC: Receive Data Bit[1]**
	RXD0_1	I		**RMII MAC: Receive Data[1] Port 0** Connect to '1' if channel is not used
	-	I	PU1	**SMII MAC: Not used**
F18	RXD0	I		**MII MAC: Receive Data Bit[0]**
	RXD0_0	I		**RMII MAC: Receive Data[0] Port 0** Connect to '1' if channel is not used
	RX_0	I		**SMII MAC: Receive Data Port 0** Connect to '1' if channel is not used
B18	-	O	PP	**MII MAC: Not used**
	-	O	PP	**RMII MAC: Not used**
	TX_CLK	O	PP	**SMII MAC: Transmit Clock** 125 MHz

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C18	-	O	PP	**MII MAC: Not used**
	REFCLK_o	O	PP	**RMII MAC: Reference Clock** Derived from internal crystal: 50 MHz
	CLOCK_o	O	PP	**SMII MAC: Reference Clock** Derived from internal crystal: 125 MHz
J16	-	O	PP	**MII MAC: Not used**
	-	O	PP	**RMII MAC: Not used**
	SYNC_o	O	PP	**SMII MAC: SYNC output signal** Used for cascading over chip boundaries
L16	**TX_ER**	O	PP	**MII MAC: Transmit Coding Error**
	-	O	PP	**RMII MAC: Not used**
	-	O	PP	**SMII MAC: Not used**
M16	TX_EN	O	PP	**MII MAC: Transmit Enable**
	TX_EN_3	O	PP	**RMII MAC: Transmit Enable Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
N16	-	O	PP	**MII MAC: Not used**
	TX_EN_2	O	PP	**RMII MAC: Transmit Enable Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
P17	-	O	PP	**MII MAC: Not used**
	TX_EN_1	O	PP	**RMII MAC: Transmit Enable Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
E17	-	O	PP	**MII MAC: Not used**
	TX_EN_0	O	PP	**RMII MAC: Transmit Enable Port 0** Do not connect if channel is not used
	TX_SYNC	O	PP	**SMII MAC: Transmit Sync**

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
T17	MDC	I/O	PP	**MII MAC: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
	MDC	I/O	PP	**RMII MAC: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
	MDC	I/O	PP	**SMII MAC: Management Data Clock** MDC input when MDIO slave (pin MDIOSLAVE==ONE) MDC output when MDIO master (pin MDIOSLAVE==ZERO) Connect to '1' if not used in MDIO slave mode
A16	MDIO	I/O	TS	**MII MAC: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
	MDIO	I/O	TS	**RMII MAC: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
	MDIO	I/O	TS	**SMII MAC: Management Data Input/Output** Connect to '1' if not used in MDIO slave mode
N17	-	O	PP	**MII MAC: Not used**
	TXD3_1	O	PP	**RMII MAC: Transmit Data[1] Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
M17	-	O	PP	**MII MAC: Not used**
	TXD3_0	O	PP	**RMII MAC: Transmit Data[0] Port 3** Do not connect for SOCRATES™-1e or if channel is not used
	TX_3	O	PP	**SMII MAC: Transmit Data Port 3** Do not connect for SOCRATES™-1e or if channel is not used
L17	-	O	PP	**MII MAC: Not used**
	TXD2_1	O	PP	**RMII MAC: Transmit Data[1] Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
K17	-	O	PP	**MII MAC: Not used**
	TXD2_0	O	PP	**RMII MAC: Transmit Data[0] Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	TX_2	O	PP	**SMII MAC: Transmit Data Port 2** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used

Table 14 I/O Signals, System Interface 1 (Parallel Data Interface), MAC Mode (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
J17	TXD3	O	PP	**MII MAC: Transmit Data[3]**
	TXD1_1	O	PP	**RMII MAC: Transmit Data[1] Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	-	O	PP	**SMII MAC: Not used**
H17	TXD2	O	PP	**MII MAC: Transmit Data[2]**
	TXD1_0	O	PP	**RMII MAC: Transmit Data[0] Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
	TX_1	O	PP	**SMII MAC: Transmit Data Port 1** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G17	TXD1	O	PP	**MII MAC: Transmit Data[1]**
	TXD0_1	O	PP	**RMII MAC: Transmit Data[1] Port 0** Do not connect if channel is not used
	-	O	PP	**SMII MAC: Not used**
F17	TXD0	O	PP	**MII MAC: Transmit Data[0]**
	TXD0_0	O	PP	**RMII MAC: Transmit Data[0] Port 0** Do not connect if channel is not used
	TX_0	O	PP	**SMII MAC: Transmit Data Port 0** Do not connect if channel is not used
C16	x_RXR_3_4	I		**MII MAC: xTALK; Rx Reference Signal[4], Channel 3**
	x_RXR_3_4	I		**RMII MAC: xTALK; Rx Reference Signal[4], Channel 3**
	x_RXR_3_4	I		**SMII MAC: xTALK; Rx Reference Signal[4], Channel 3**
C14	x_RXR_3_3	I		**MII MAC: xTALK; Rx Reference Signal[3], Channel 3**
	x_RXR_3_3	I		**RMII MAC: xTALK; Rx Reference Signal[3], Channel 3**
	x_RXR_3_3	I		**SMII MAC: xTALK; Rx Reference Signal[3], Channel 3**
T14	x_RXR_3_2	I		**MII MAC: xTALK; Rx Reference Signal[2], Channel 3**
	x_RXR_3_2	I		**RMII MAC: xTALK; Rx Reference Signal[2], Channel 3**
	x_RXR_3_2	I		**SMII MAC: xTALK; Rx Reference Signal[2], Channel 3**
V15	x_RXR_3_1	I		**MII MAC: xTALK; Rx Reference Signal[1], Channel 3**
	x_RXR_3_1	I		**RMII MAC: xTALK; Rx Reference Signal[1], Channel 3**
	x_RXR_3_1	I		**SMII MAC: xTALK; Rx Reference Signal[1], Channel 3**
V14	x_RXR_3_0	I		**MII MAC: xTALK; Rx Reference Signal[0], Channel 3**
	x_RXR_3_0	I		**RMII MAC: xTALK; Rx Reference Signal[0], Channel 3**
	x_RXR_3_0	I		**SMII MAC: xTALK; Rx Reference Signal[0], Channel 3**

Table 15 Not Used Pins MAC Mode,

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
A15, B15, T12	PU	I	PU1	**Not used**
B13, K16	- (not used)	O	PP	**Not used**

2.2.6 System Interface 2

Pins which are not used in the packet application (description of the functionality: Not Used) may either be left open or be connectet to ’1’.

Table 16 I/O Signals, System Interface 2 (Serial Data Interface (SDI) Channel 0)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
V17	**TFSC_0**	I		**Normal: Transmit Frame Clock, Channel 0** Programmable Frequency (125 µs).
	CSP	I		**PacketOnly: Common Sync Puls** Used for Parallel Data Interface (Utopia/POS-PHY or xMII) to synchronize the SHDSL link when Serial Data Interface is not in use Programmable Frequency (6 ms, 125 µs) If this synchronization is not needed, this pin has to be tied to ’0’.
V16	**TCLK_0**	I		**Normal: Transmit Data Clock, Channel 0** Programmable Frequency.
	CCLK	I		**PacketOnly: Common Clock** Used for Parallel Data Interface (Utopia/POS-PHY or xMII) to synchronize the SHDSL link when Serial Data Interface is not in use Programmable Frequency If this synchronization is not needed, this pin has to be tied to ’0’
U18	**DIN_0**	I		**Normal: Data In, Channel 0**
	-	I	PU1	**PacketOnly: Not used**
T16	**RFSC_0**	O	PP	**Normal: Receive Frame Clock, Channel 0** Programmable Frequency (6 ms, 125 µs). Buffer: C
	-	I	PU1	**PacketOnly: Not used**
U16	**RCLK_0**	O	PP	**Normal: Receive Data Clock, Channel 0** Programmable Frequency. Buffer: C
	-	I	PU1	**PacketOnly: Not used**
V18	**DOUT_0**	O	TS	**Normal: Data Out, Channel 0** Buffer: C
	-	O	TS	**PacketOnly: Not used**

2.2.7 JTAG Interface

Table 17 I/O Signals, JTAG Interface

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B14	TDO	O	PP	**Test Data Output** Buffer: C
B12	TDI	I	PU1	**Test Data Input**
A14	TCK	I	PU1	**Test Clock** If JTAG interface is hold in reset state, the clock input does not need to be driven
A12	TMS	I	PU1	**Test Mode Select**
A13	$\overline{\text{TRST}}$	I	PU1	**Test Reset** active low, during normal operation of the chip this pin must be connected to '0'

2.2.8 Analog Interface

Table 18 I/O Signals of Analog Part

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
U10	XTAL1	I		**Crystal In (20.48 MHz/40.96 MHz)** *Note: see pins pins TP1, TP2 for frequency selection*
V10	XTAL2	O		**Crystal Out**
T10	CLKOUT	O		**Clock Out (20.48 MHz/40.96 MHz)** Clock derived from oscillator
Analog Front End, Channel 0				
V4	CREF1_0	AI/O		**VDD Reference Voltage** Do not connect if channel is not used
V5	CREF2_0	AI/O		**VSS Reference Voltage** Do not connect if channel is not used
U4	RTIP_0	AI		**Receive Tip input** To be differential routed with RRING_0 Connect to V_{SSA} if channel is not used
T4	RRING_0	AI		**Receive Ring input** To be differential routed with RTIP_0 Connect to V_{SSA} if channel is not used
R1	VCM_0	AI/O		**Midband for Line Driver** Do not connect if channel is not used
U1	TTIP1_0	AO		**Transmit Tip, Line driver 1 output** To be differential routed with TTIP2_0, TRING1_0 and TRING2_0 Do not connect if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
U2	TTIP2_0	AO		**Transmit Tip, Line driver 2 output** To be differential routed with TTIP1_0, TRING1_0 and TRING2_0 Do not connect if channel is not used
T1	TRING1_0	AO		**Transmit Ring, Line driver 1 output** To be differential routed with TTIP1_0, TTIP2_0 and TRING2_0 Do not connect if channel is not used
T2	TRING2_0	AO		**Transmit Ring, Line driver 2 output** To be differential routed with TTIP1_0, TTIP2_0 and TRING1_0 Do not connect if channel is not used
U3	SZTIP_0	AI		**Synthesized Impedance Input Tip** To be differential routed with SZRING_0 Connect to V_{SSA} if channel is not used
T3	SZRING_0	AI		**Synthesized Impedance Input Ring** To be differential routed with SZTIP_0 Connect to V_{SSA} if channel is not used
T6	FBRING_0	AI		**Line Driver Feed-Back Input Ring** To be differential routed with FBTIP_0 Connect to V_{SSA} if channel is not used
U6	FBTIP_0	AI		**Line Driver Feed-Back Input Tip** To be differential routed with FBRING_0 Connect to V_{SSA} if channel is not used
U5	HYB1_0	AI/O		**Hybrid (Connect to hybrid circuit)** To be differential routed with HYB2_0 Do not connect if channel is not used
T5	HYB2_0	AI/O		**Hybrid (Connect to hybrid circuit** To be differential routed with HYB1_0 Do not connect if channel is not used
U7	TAO_0	AO		**Analog output** For test purposes only. Has to be left open during normal operation Do not connect if channel is not used
U8	GUARD_0	AI		**Guard** Has to be connected to quiet V_{SS} To be also connected if channel is not used
Analog Front End, Channel1 (Channel not supported by SOCRATES™-1e and SOCRATES™-2e)				
V7	CREF1_1	AI/O		**VDD Reference Voltage** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
V8	CREF2_1	AI/O		**VSS Reference Voltage** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
N4	RTIP_1	AI		**Receive Tip input** To be differential routed with RRING_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M4	RRING_1	AI		**Receive Ring input** To be differential routed with RTIP_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
L1	VCM_1	AI/O		**Midband for Line Driver** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
N1	TTIP1_1	AO		**Transmit Tip, Line driver 1 output** To be differential routed with TTIP2_1, TRING1_1 and TRING2_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
N2	TTIP2_1	AO		**Transmit Tip, Line driver 2 output** To be differential routed with TTIP1_1, TRING1_1 and TRING2_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M1	TRING1_1	AO		**Transmit Ring, Line driver 1 output** To be differential routed with TTIP1_1, TTIP2_1 and TRING2_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M2	TRING2_1	AO		**Transmit Ring, Line driver 2 output** To be differential routed with TTIP1_1, TTIP2_1 and TRING1_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
N3	SZTIP_1	AI		**Synthesized Impedance Input Tip** To be differential routed with SZRING_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M3	SZRING_1	AI		**Synthesized Impedance Input Ring** To be differential routed with SZTIP_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M6	FBRING_1	AI		**Line Driver Feed-Back Input Ring** To be differential routed with FBTIP_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
N6	FBTIP_1	AI		**Line Driver Feed-Back Input Tip** To be differential routed with FBRING_1 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
N5	HYB1_1	AI/O		**Hybrid (Connect to hybrid circuit)** To be differential routed with HYB2_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
M5	HYB2_1	AI/O		**Hybrid (Connect to hybrid circuit** To be differential routed with HYB1_1 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
T7	TAO_1	AO		**Analog output** For test purposes only. Has to be left open during normal operation Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
T8	GUARD_1	AI		**Guard** Has to be connected to quiet V_{SS} To be also connected for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
Analog Front End, Channel 2 (not supported by SOCRATES™-1e and SOCRATES™-2e)				
A7	CREF1_2	AI/O		**VDD Reference Voltage** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
A8	CREF2_2	AI/O		**VSS Reference Voltage** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
F4	RTIP_2	AI		**Receive Tip input** To be differential routed with RRING_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G4	RRING_2	AI		**Receive Ring input** To be differential routed with RTIP_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
H1	VCM_2	AI/O		**Midband for Line Driver** Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
F1	TTIP1_2	AO		**Transmit Tip, Line driver 1 output** To be differential routed with TTIP2_2, TRING1_2 and TRING2_2 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F2	TTIP2_2	AO		**Transmit Tip, Line driver 2 output** To be differential routed with TTIP1_2, TRING1_2 and TRING2_2 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G1	TRING1_2	AO		**Transmit Ring, Line driver 1 output** To be differential routed with TTIP1_2, TTIP2_2 and TRING2_2 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G2	TRING2_2	AO		**Transmit Ring, Line driver 2 output** To be differential routed with TTIP1_2, TTIP2_2 and TRING1_2 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
F3	SZTIP_2	AI		**Synthesized Impedance Input Tip** To be differential routed with SZRING_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G3	SZRING_2	AI		**Synthesized Impedance Input Ring** To be differential routed with SZTIP_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G6	FBRING_2	AI		**Line Driver Feed-Back Input Ring** To be differential routed with FBTIP_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
F6	FBTIP_2	AI		**Line Driver Feed-Back Input Tip** To be differential routed with FBRING_2 Connect to V_{SSA} for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
F5	HYB1_2	AI/O		**Hybrid (Connect to hybrid circuit)** To be differential routed with HYB2_0 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
G5	HYB2_2	AI/O		**Hybrid (Connect to hybrid circuit** To be differential routed with HYB1_2 Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
C7	TAO_2	AO		**Analog output** For test purposes only. Has to be left open during normal operation Do not connect for SOCRATES™-1e and SOCRATES™-2e or if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C8	GUARD_2	AI		**Guard** Has to be connected to quiet V_{SS} To be also connected for SOCRATES™-1e and SOCRATES™-2e or if channel is not used
Analog Front End, Channel 3 (Channel not supported by SOCRATES™-1e)				
A4	CREF1_3	AI/O		**VDD Reference Voltage** Do not connect for SOCRATES™-1e or if channel is not used
A5	CREF2_3	AI/O		**VSS Reference Voltage** Do not connect for SOCRATES™-1e or if channel is not used
B4	RTIP_3	AI		**Receive Tip input** To be differential routed with RRING_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
C4	RRING_3	AI		**Receive Ring input** To be differential routed with RTIP_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
D1	VCM_3	AI/O		**Midband for Line Driver** Do not connect for SOCRATES™-1e or if channel is not used
B1	TTIP1_3	AO		**Transmit Tip, Line driver 1 output** To be differential routed with TTIP2_3, TRING1_3 and TRING2_3 Do not connect for SOCRATES™-1e or if channel is not used
B2	TTIP2_3	AO		**Transmit Tip, Line driver 2 output** To be differential routed with TTIP1_3, TRING1_3 and TRING2_3 Do not connect for SOCRATES™-1e or if channel is not used
C1	TRING1_3	AO		**Transmit Ring, Line driver 1 output** To be differential routed with TTIP1_3, TTIP2_3 and TRING2_3 Do not connect for SOCRATES™-1e or if channel is not used
C2	TRING2_3	AO		**Transmit Ring, Line driver 2 output** To be differential routed with TTIP1_3, TTIP2_3 and TRING1_3 Do not connect for SOCRATES™-1e or if channel is not used

Table 18 I/O Signals of Analog Part (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B3	SZTIP_3	AI		**Synthesized Impedance Input Tip** To be differential routed with SZRING_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
C3	SZRING_3	AI		**Synthesized Impedance Input Ring** To be differential routed with SZTIP_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
C6	FBRING_3	AI		**Line Driver Feed-Back Input Ring** To be differential routed with FBTIP_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
B6	FBTIP_3	AI		**Line Driver Feed-Back Input Tip** To be differential routed with FBRING_3 Connect to V_{SSA} for SOCRATES™-1e or if channel is not used
B5	HYB1_3	AI/O		**Hybrid (Connect to hybrid circuit)** To be differential routed with HYB2_3 Do not connect for SOCRATES™-1e or if channel is not used
C5	HYB2_3	AI/O		**Hybrid (Connect to hybrid circuit** To be differential routed with HYB1_3 Do not connect for SOCRATES™-1e or if channel is not used
B7	TAO_3	AO		**Analog output** For test purposes only. Has to be left open during normal operation Do not connect for SOCRATES™-1e or if channel is not used
B8	GUARD_3	AI		**Guard** Has to be connected to quiet V_{SS} To be also connected for SOCRATES™-1e or if channel is not used

2.2.9 Power Supply and Not Connected

Table 19 Power Supply

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F12, G12, H12, J12, K12, L12, M12, N12	VDDP	PWR	–	**Supply Voltage for the Pads** 3.3 V
F14, G14, H14, J14, K14, L14, M14, N14	VDDD	PWR	–	**Supply Voltage for the Digital Part** 1.5 V
F11, G11, H11, J11, K11, L11, M11, N11, F15, G15, H15, J15, K15, L15, M15, N15	VSSD	GND	–	**Ground for the Pads and for the Digital Part**

Table 19 Power Supply (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
A2, D2, D4, D6, D8, E2, E4, E6, E8, F8, G8, H2, H4, H6, H8, J2, J4, J6, J8, K2, K4, K6, K8, L2, L4, L6, L8, M8, N8, P2, P4, P6, P8, R2, R4, R6, R8, V2	VDDA	PWR	–	**Supply Voltage for the Analog Part** 3.3 V
A1, A3, A6, A9, B9, C9, D3, D5, D7, D9, E1, E3, E5, E7, E9, F7, F9, G7, G9, H3, H5, H7, H9, J1, J3, J5, J7, J9, K1, K3, K5, K7, K9, L3, L5, L7, L9, M7, M9, N7, N9, P1, P3, P5, P7, P9, R3, R5, R7, R9, T9, U9, V1, V3, V6, V9	VSSA	GND	–	**Ground for the Analog Part**

2.3 Abbreviations

The following abbreviations are used in **Ball Definitions and Functions**.

Table 20 Abbreviations for Pin Type

Abbreviations	Description
I	Standard input-only pin. Digital levels.
O	Output. Digital levels.
I/O	I/O is a bidirectional input/output signal.
AI	Input. Analog levels.
AO	Output. Analog levels.
AI/O	Input or Output. Analog levels.

Table 20 Abbreviations for Pin Type (cont'd)

Abbreviations	Description
PWR	Power
GND	Ground
MCL	Must be connected to Low (JEDEC Standard)
MCH	Must be connected to High (JEDEC Standard)
NU	Not Usable (JEDEC Standard) *Note: Pad disabled*
NC	Not Connected (JEDEC Standard)

Table 21 Abbreviations for Buffer Type

Abbreviations	Description
Z	High impedance
PU1	Pull up *Note: Ball can either be left open or connected to High*
PD1	Pull down *Note: Ball can either be left open or connected to Low*
TS	Tristate capability: The corresponding pin has 3 operational states: Low, high and high-impedance.
OD	Open Drain. The corresponding pin has 2 operational states, active low and tristate, and allows multiple devices to share as a wire-OR. An external pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
OC	Open Collector
PP	Push-Pull. The corresponding pin has 2 operational states: Active-low and active-high (identical to output with no type attribute).
OD/PP	Open-Drain or Push-Pull. The corresponding pin can be configured either as an output with the OD attribute or as an output with the PP attribute.
ST	Schmitt-Trigger characteristics
TTL	TTL characteristics
Buffer A, B, C	Driver Strength of output buffer, see **Chapter 7.5**

3 System Integration

Since the hardware resources are enabled by firmware, refer to the Firmware Release Notes for the supported feature set.

3.1 Reference Model

The reference model for SHDSL used in this document is according to ITU-T G.991.2, ITU-T G.991.2 (2004), and ETSI TS 101 524 V 1.2.1.

The reference model for EFM used in this document is according to IEEE 802.3-2004.

Figure 17 depicts the reference model for SHDSL and **Figure 18** for EFM.

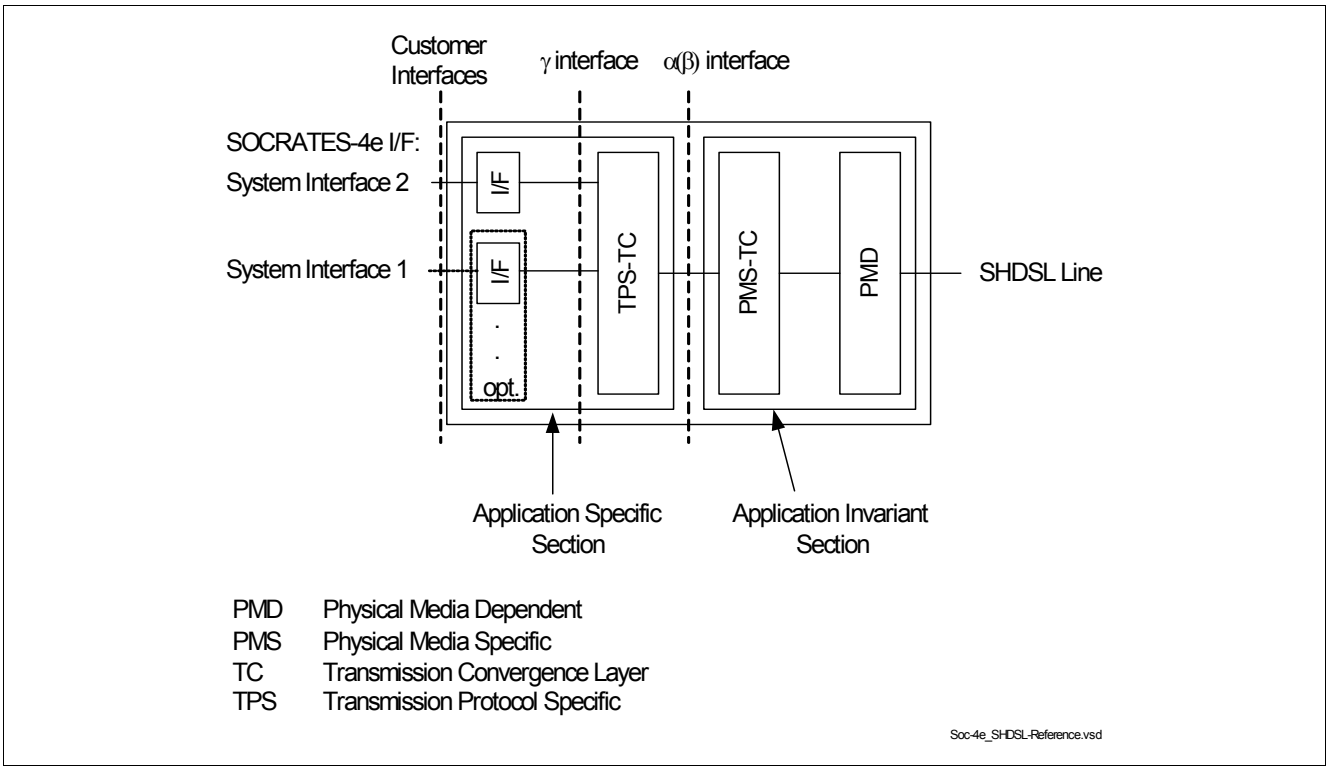


Figure 17 SHDSL Reference Model

System Interface 1 refers to the packet data interface, System Interface 2 to the TDM interface (see also **Chapter 1.5**).

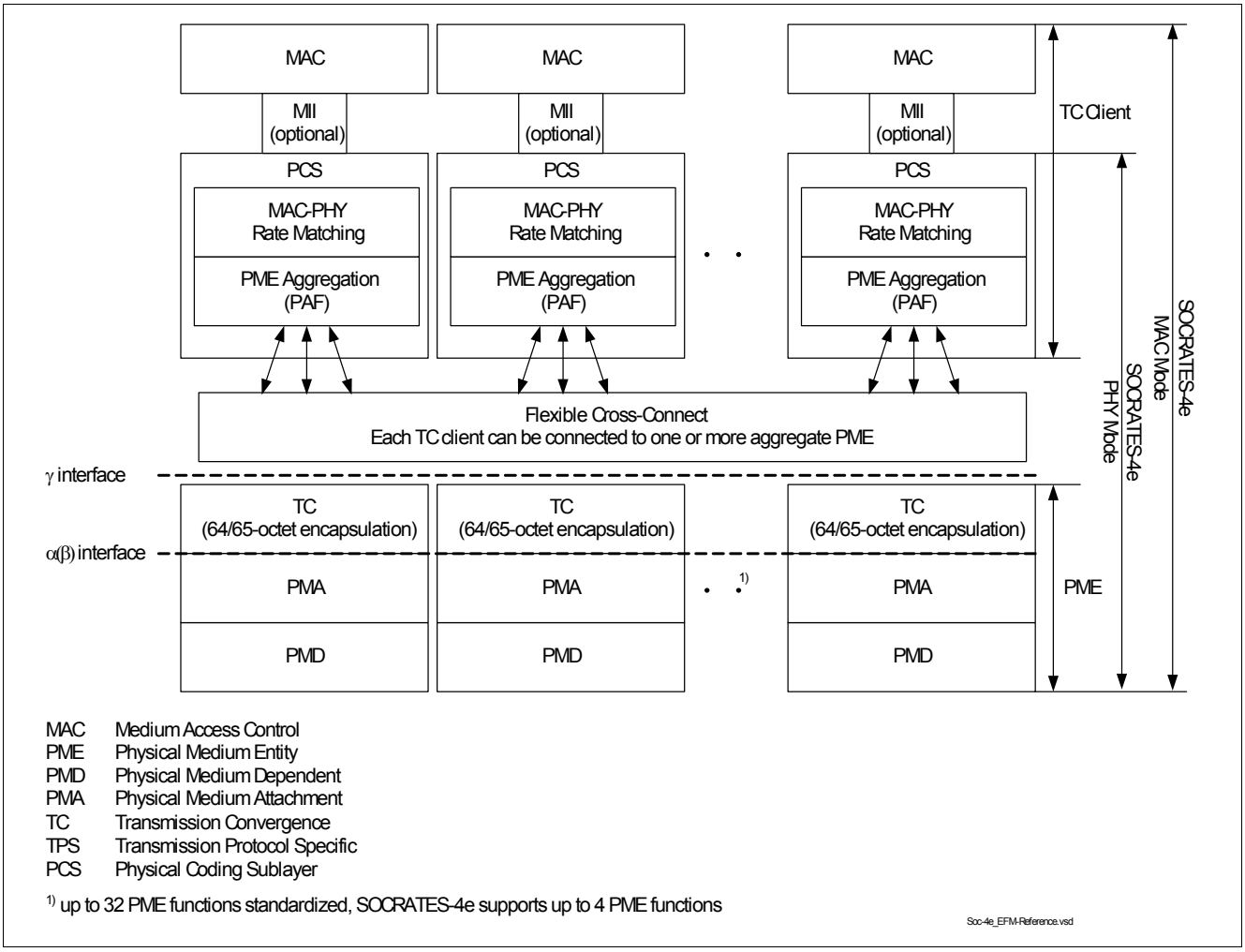


Figure 18 EFM Reference Model

3.2 Clocking Scheme

According to ITU-T G.991.2 (2004) there are 4 different clocking modes standardized for the SHDSL link, which are all supported by the SOCRATES™-4e. The following subclauses summarize these clock modes, describe the clock accuracy to be applied to the SOCRATES™-4e and summarize the clock modes supported by the SOCRATES™-4e in more detail.

3.2.1 Reference Clock

The clock reference options described in this chapter apply to the reference clock architecture according to the ITU-T G.991.2 (2004) standard and are described in more detail in the following chapters.

Figure 19 shows a simplified SHDSL reference model and **Table 22** lists the timing modes standardized in ITU-T G.991.2 (2004).

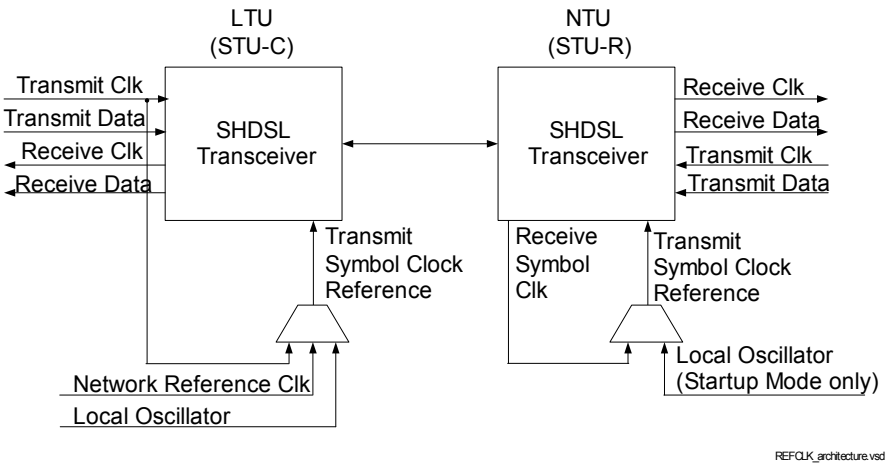


Figure 19 Reference Clock Architecture Customer Interface Clock

Table 22 Clock Synchronization Configuration

Timing Mode#	STU-C Symbol Clock Reference	STU-R Symbol Clock Reference	Example Application	Mode
1	Local oscillator	Received symbol clock	'Classic' HDSL	Plesiochronous
2	Network reference clock	Received symbol clock	'Classic' HDSL with embedded timing reference	Plesiochronous with timing reference
3a	Transmit data clock or network reference clock	Received symbol clock	Main application	Synchronous
3b	Transmit data clock	Received symbol clock	Synchronous downstream and bit-stuffed upstream	**Downstream** synchronous **Upstream** plesiochronous

3.2.1.1 Reference Clock Frequencies

For the reference clock pin a wide range of frequencies can be programmed. **Table 23** shows the possible reference clock frequencies when pin REFCLK_# is programmed to input and **Table 24** for pin REFCLK_# programmed as output.

Table 23 Supported Input Frequencies of Pin REFCLK_#

Frequency	Minimum Low/High Phase
8 kHz	One 60 MHz period
1536 kHz	One 60 MHz period
1544 kHz	One 60 MHz period
2048 kHz	One 60 MHz period
4096 kHz	One 60 MHz period
8192 kHz	One 60 MHz period
15.36 MHz	One 60 MHz period
20.48 MHz	One 60 MHz period

Table 24 Supported Output Frequencies of Pin REFCLK_#

Frequency[1)]	Duty Cycle	Clock Source
8 kHz	40-60 to 60-40	Rx Data Clk, Rx Symbol Clk or 20.48 MHz
1536 kHz	40-60 to 60-40	Rx Data Clk
1544 kHz	40-60 to 60-40	Rx Data Clk
2048 kHz	40-60 to 60-40	Rx Data Clk
4096 kHz	40-60 to 60-40	Rx Data Clk
8192 kHz	40-60 to 60-40	Rx Data Clk
15.36 MHz	40-60 to 60-40	Rx Data Clk
19.44 MHz	40-60 to 60-40	Rx Data Clk
20.48 MHz	40-60 to 60-40	Rx Data Clk
24.576 MHz	40-60 to 60-40	Rx Data Clk

1) An intrinsic jitter of ± 8.14 ns

3.2.2 Frequency and Jitter Tolerance

The SOCRATES™-4e tolerates input frequency differences of up to ± 150 ppm between Transmit Clk and Local Oscillator (refer also to **Figure 19**).

The SOCRATES™-4e fulfils the input jitter tolerance requirements and jitter transfer functions according to the following standards:

1. ITU-T I.431 for 1544 kbit/s and 2048 kbit/s (ISDN Primary Rate User-Network)
2. ITU-T G.823 for 2048 kbit/s (PDH)
3. ITU-T G.824 for 1544 kbit/s (PDH)
4. ITU-T G.736 for 2048 kbit/s (Digital Multiplex Equipment)

3.2.3 Supported Clocking Modes

The SOCRATES™-4e supports all clocking modes standardized in ITU-T G.991.2 (2004) which are summarized in **Chapter 3.2.1**.

The pins REFCLK_# represent the Symbol Clock Reference for each SHDSL channel as depicted in **Figure 19** in STU-C application. Furthermore the pins XTAL, CSP (in packet mode only) and TFSC_0 are relevant for the clocking. **Table 25** shows the different clocking modes and the references for that mode. In STU_C application the pin REFCLK_# is always configured as input, in STU-R application the pin STU-R is always an output. In case the REFCLK_# in STU-C application is the reference for several SHDSL line, the REFCLK_# inputs need to be connected externally.

Table 25 Clocking Modes

Timing Mode#	Symbol Reference Clock	Application	Clocking Reference	Accuracy of Reference	XTAL accuracy
STU-C Side					
1	Local Oscillator	Packet	XTAL[1)]	+/- 32 ppm	+/-32 ppm
		TDM	XTAL (REFCLK input used for SDI synchronization)	+/- 32 ppm	+/-32 ppm

Table 25 Clocking Modes (cont'd)

Timing Mode#	Symbol Reference Clock	Application	Clocking Reference	Accuracy of Reference	XTAL accuracy
2	Network reference clock	Packet	REFCLK	+/- 32 ppm	+/- 60 ppm
		TDM	REFCLK(TDM clock has to have same source as REFCLK, jitter to REFCLK allowed)	+/- 32 ppm	+/- 60 ppm
3a	Transmit data clock	Packet	XTAL[1)]	+/- 32 ppm	+/- 32 ppm
		Packet	CSP[2)]	+/- 32 ppm	+/- 60 ppm
		Packet	REFLCK[3)]	+/- 32 ppm	
		TDM	TFSC_0[2)]	+/- 32 ppm	
		TDM	TFSC_0 and REFCLK (externally connected)[3)]	+/- 32 ppm	
3b	Transmit data clock	Packet	XTAL[1)]	+/- 32 ppm	+/- 32 ppm
		Packet	CSP[2)]	+/- 32 ppm	+/- 60 ppm
		Packet	REFLCK[3)]	+/- 32 ppm	
		TDM	TFSC_0[2)]	+/- 32 ppm	
		TDM	TFSC_0 and REFCLK (externally connected)[3)]	+/- 32 ppm	
STU-R Side					
1	Received symbol clock	Packet/TDM	received symbol clock	+/- 32 ppm	+/- 60 ppm
2	Received symbol clock				
3a	Received symbol clock				
3b	Received symbol clock				

1) The REFCLK and the CSP input is not relevant in that application, it needs to be tied to either '0' or '1'.

2) The REFCLK input is not relevant in that application, it needs to be tied to eiher '0' or '1'.

3) This configuration is prefered in case of jitter sensitve applications. The CSP input is not relevant in packet only application, it needs to be tied to either '0' or '1'. In TDM or Dual Bearer application (combination of packet and TDM data) the REFCLK and TFSC_0 need to be connected to the timing reference.

Note that in all EFM modi according to IEEE 802.3-2004 only clock mode 3a is supported. In case the TDM interface is used (dual bearer application), the clocking provided at this interface determines the clocking of both bearers.

In case of a combination of single pair and m-pair the lines forming the m-pair group need to be configured first.

3.3 Packet and Ethernet Modes

For CO applications as well as CPE applications using the System interface 1 in xMII mode the SOCRATES™-4e can either work in PHY or in MAC mode. The differences of these modes are described in the following subclauses.

3.3.1 SOCRATES™-4e in PHY Mode

In PHY mode the SOCRATES™-4e works as pure Ethernet (EFM) PHY as standardized in clause 61 and 63 of IEEE 802.3-2004. To terminate the EFM connection a MAC has to be connected externally to the xMII. **Figure 20** depicts the building blocks of the SOCRATES™-4e used for the PHY mode. As alternative to the 64/65-octet TC as standardized in IEEE 802.3-2004 the PTM TC as standardized at ITU-T G.991.2 (2004) can be used.

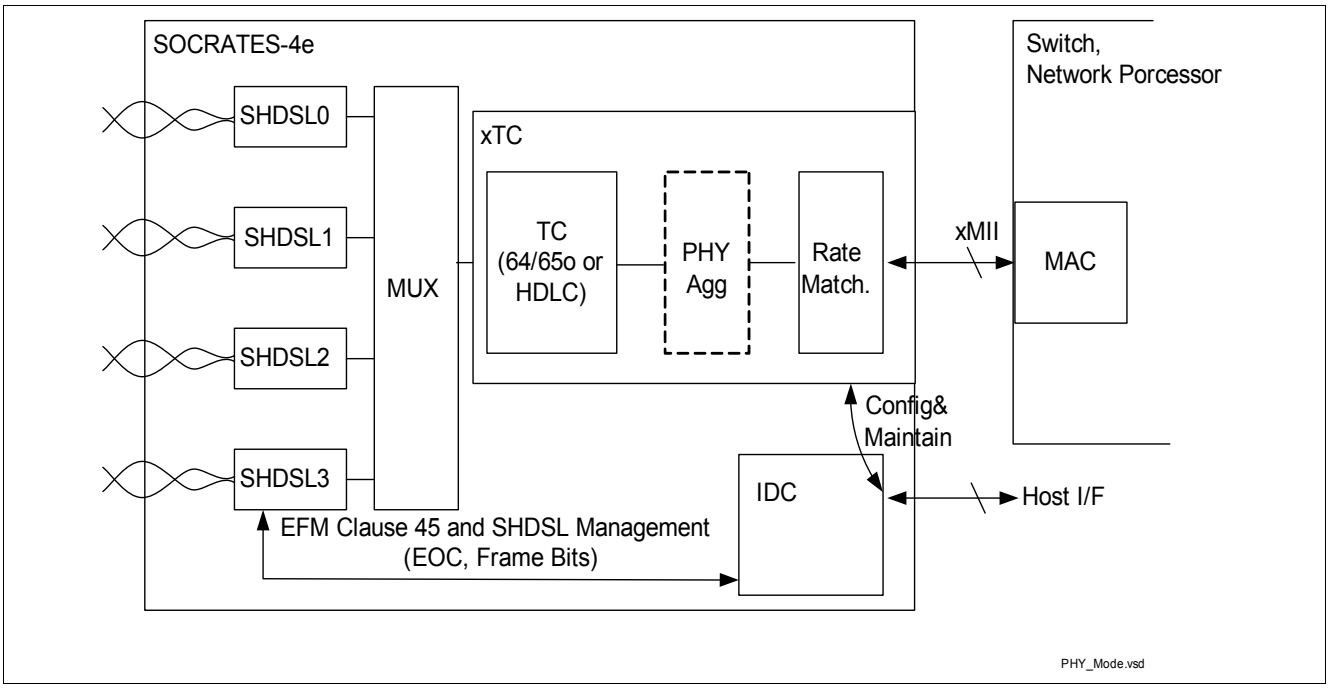


Figure 20 PHY Mode

3.3.2 SOCRATES™-4e in MAC Mode

In MAC mode the SOCRATES™-4e terminates the EFM connection, an external Ethernet PHY can be directly connected. Programmable filters allow access of the IDC to the payload of Ethernet packets of the SHDSL link, i.e. OAM and/or other Ethernet packets can be extracted/inserted to/from the Host interface from/to the local loop via the IDC. All frames for which the filter settings do not match are bypassed in direction towards MII.

Programmable filters, which are different from the ones working towards the SHDSL link, allow access of the IDC to the payload of Ethernet packets of MII, i.e. OAM and/or other Ethernet packets can be extracted/inserted to/from the Host interface from/to the MII via the IDC. All frames for which the filter settings do not match are bypassed in direction towards the SHDSL link.

Figure 21 depicts the building blocks of the SOCRATES™-4e used for the MAC mode. As alternative to the 64/65-octet TC as standardized in IEEE 802.3-2004 the PTM TC as standardized at ITU-T G.991.2 (2004) can be used.

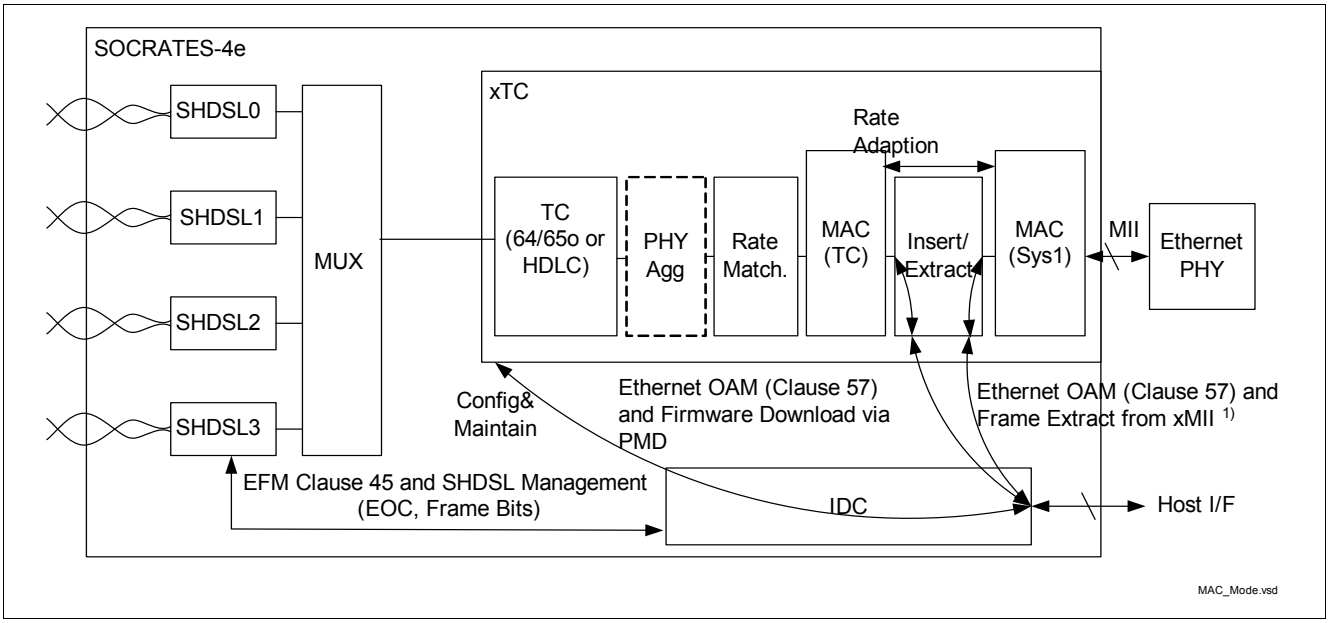


Figure 21 MAC Mode

3.4 ATM Modes

The SOCRATES™-4e also supports plain ATM application. In this case the System Interface 1 has to be operated either in UTOPIA or in POS-PHY mode. **Figure 22** shows the building blocks of the SOCRATES™-4e in ATM mode.

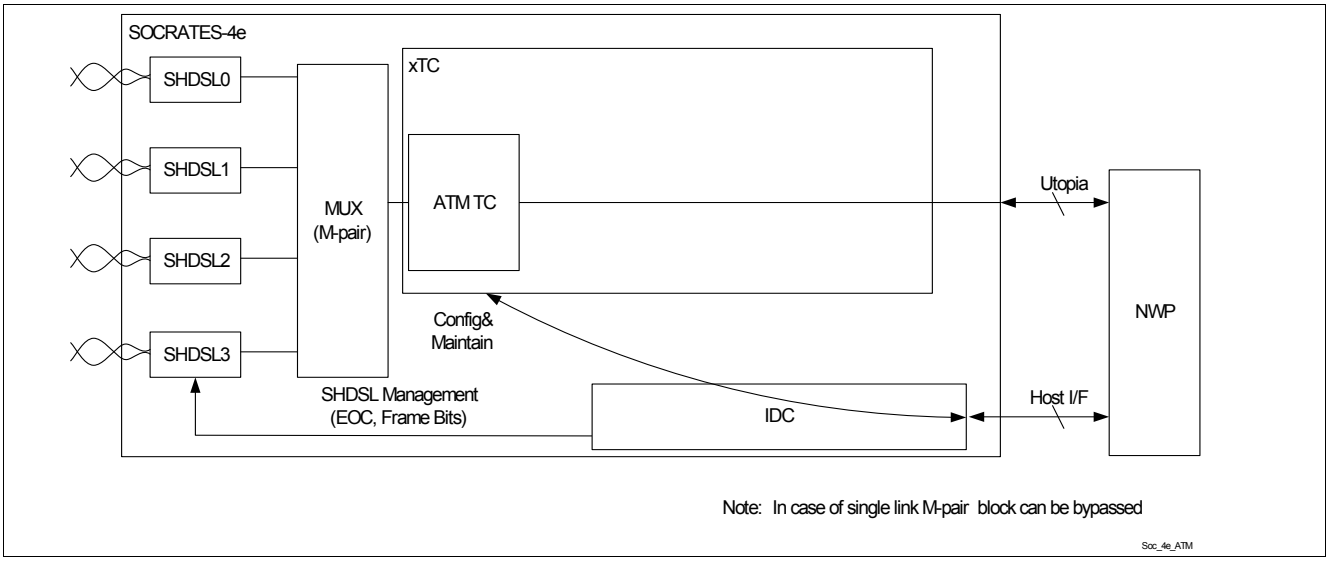


Figure 22 ATM Modes

3.5 Interworking

The SOCRATES™-4e is suitable for interworking applications on CO and CPE side, i.e. interworking in both directions is supported.

Figure 23 depicts the building blocks for an interworking scenario with ATM on System Interface 1 and Ethernet on the local loop. This scenario is related to applications where the interworking functionality is typically located in the CO, e.g. EFM is provided out of an ATM based DLSAM/network.

Figure 24 depicts the building blocks for an interworking scenario with ATM on the local loop and Ethernet on System Interface 1. This scenario is related to applications where the interworking functionality is typically located in the CPE, hence low featured network processors (NWP's) can be used for BOM optimized CPE designs.

ATM on System Interface 1

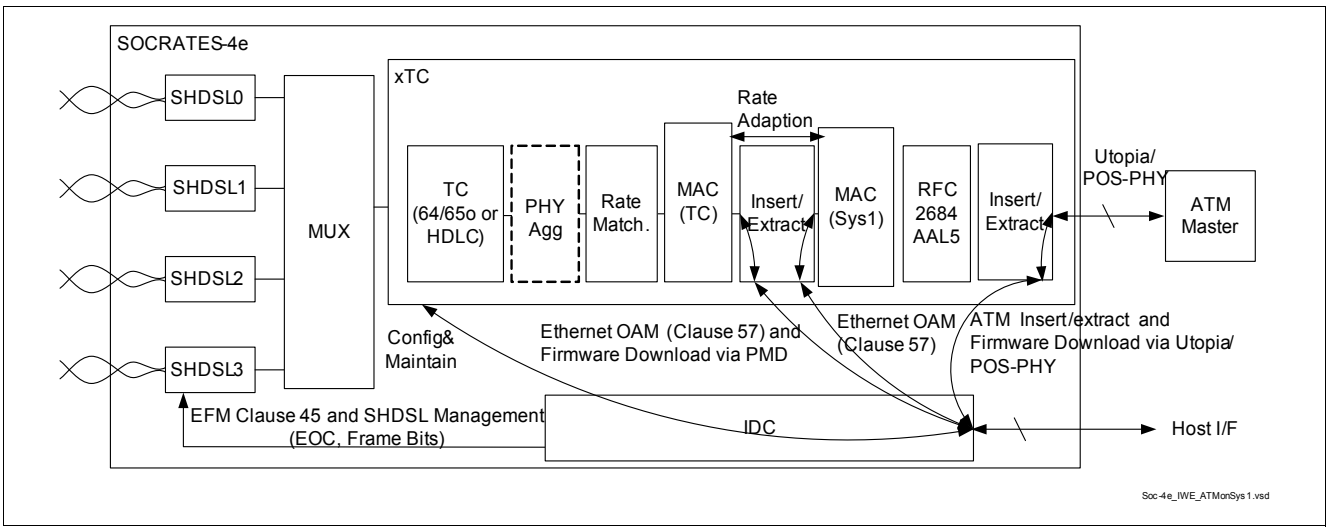


Figure 23 Interworking, ATM on System Interface 1

ATM on Line Side

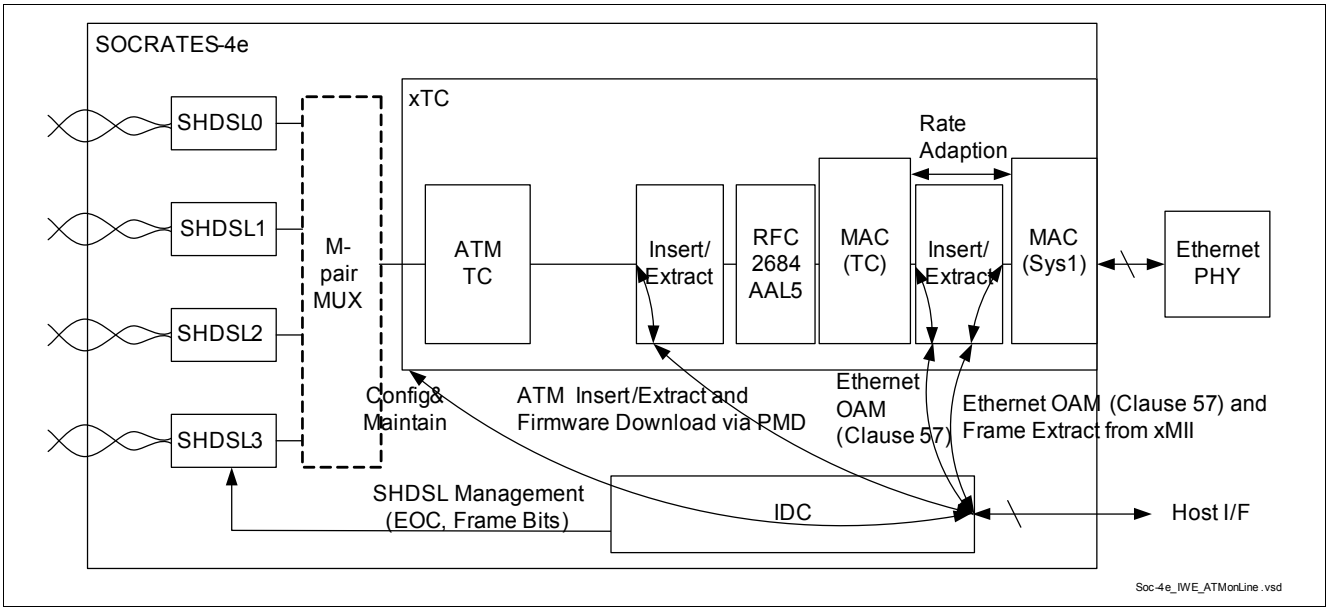


Figure 24 Interworking, ATM on Line Side

3.6 Dual Bearer Mode

In dual bearer mode the data streams of System Interface 1 and System Interface 2 are combined on the SHDSL link. The first bits of one SHDSL payload sub-block are transferred to/from the System Interface 2 and the remaining bits of one SHDSL payload sub-block are transferred to/from the System Interface 1.

3.7 Loop Backs

Figure 25 shows the loops of the SOCRATES™-4e, which are independently implemented for each System Interface port and SHDSL channel.

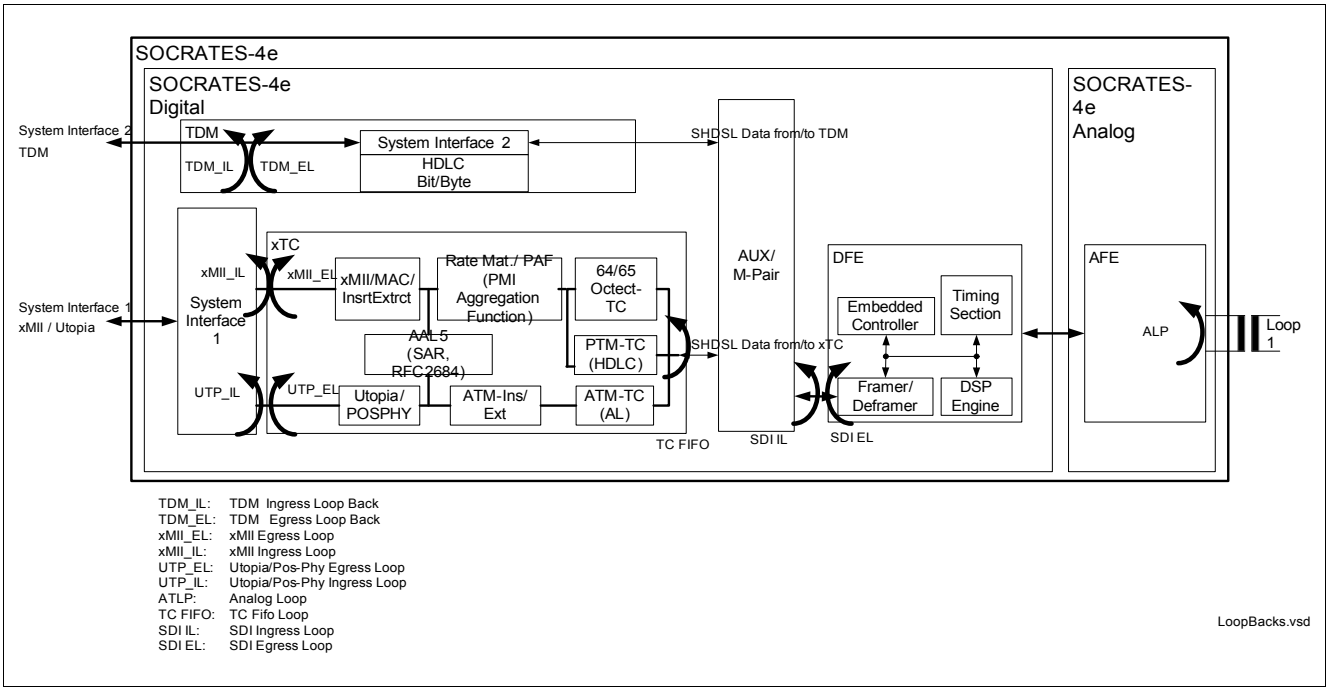


Figure 25 Loop Backs of one System Interface Port and one SHDSL Channel

TDM Egress Loop Back (TDM_EL)

The TDM egress loop back loops the TDM data received from the line interface back to the line interface. The looped back data is visible on the TDM interface, i.e. the TDM egress loop back is transparent.

This loop back simplifies the analysis of an established data connection and is closed inside the TDM block as close as possible to the TDM pins.

The EOC channel and System Interface 1 data remain untouched by this loop back, i.e. EOC messages can be received from and transmitted to the counterpart and data can be transferred in the dual bearer mode via System Interface 1.

TDM Transparent Ingress Loop Back (TDM_TIL)

The TDM transparent ingress loop back loops the TDM data received from the TDM interface back to the TDM output port. The looped back data is visible on the SHDSL link, i.e. this TDM ingress loop back is transparent.

This loop back simplifies the analyses of the board connections.

TDM Non Transparent Ingress Loop Back (TDM_NTIL)

The TDM non transparent ingress loop back loops the TDM data received from the TDM interface back to the TDM output port. This loop is not aligned with respect to the RFSC pin. The looped back data is not visible on the SHDSL link, i.e. this TDM ingress loop back is non transparent.

This loop back simplifies the analyses of the board connections.

xMII Egress Loop Back (xMII_EL)

The xMII egress loop back loops the Ethernet data received from the line interface back to the line interface. The looped back data is not visible on xMII, thus the xMII egress loop back is non transparent.

This loop back simplifies the analyses of an established data connection and is closed inside the SOCRATES™-4e as close as possible to the xMII pins.

The EOC channel and System Interface 2 data remain untouched by this loop back, i.e. EOC messages can be received from and transmitted to the counterpart and data can be transferred in the dual bearer mode via System Interface 2.

xMII Ingress Loop Back (xMII_IL)

The xMII ingress loop back loops the xMII data received from the xMII interface back to the xMII interface. The looped back data is not visible on the SHDSL link, thus the xMII ingress loop back is non transparent.

This loop back simplifies the analyses of the board connections.

UTOPIA Egress Loop Back (UTP_EL)

The UTOPIA egress loop back loops the UTOPIA data received from the line interface back to the line interface.The looped back data is not visible on UTOPIA, i.e. the UTOPIA egress loop back is non transparent.

This loop back simplifies the analyses of an established data connection and is closed inside the SOCRATES™-4e as close as possible to the UTOPIA/POS-PHY pins.

The EOC channel and System Interface 2 data remain untouched by this loop back, i.e. EOC messages can be received from and transmitted to the counterpart and data can be transferred in the dual bearer mode via System Interface 2.

UTOPIA Ingress Loop Back (UTP_IL)

The UTOPIA ingress loop back loops the UTOPIA data received from the UTOPIA/POS-PHY interface back to the UTOPIA/POS-PHY interface. The looped back data is not visible on the SHDSL link, i.e. the UTOPIA ingress loop back is non transparent.

This loop back simplifies the analyses of the board connections.

Analog Loop Back (ALP)

The analog loop back loops the data received from the DSP back to the DSP. This loop back simplifies the analyses of the data path through the SOCRATES™-4e and is closed inside the analog part as close as possible to the analog pins.

To simplify the analysis of the external circuitry the data is also transmitted to the loop interface, therefore the loop is transparent.

4 Functional Description

Since the hardware resources are enabled by firmware, refer to the Firmware Release Notes for the supported feature set.

In this chapter the building blocks of the SOCRATES™-4e are described in more detail.

4.1 Functional Block Diagram

The SOCRATES™-4e consists out of 2 major building blocks, the AFE and the DFE.

The AFE block consists out of the analog front end for the 4 SHDSL channels including line drivers. It performs analog to digital, digital to analog conversion, echo suppression and delivers up to 16.8 dBm output power (16.8 dBm is used for asymmetric PSDs) for the transmit signal. The DFE includes the 4 DSP's realizing the SHDSL PMA/PMD functionality including cross talk cancelling. The xTC block includes all the TC functionality for all cell and packet data streams for 4 channels. Additionally it includes interworking functionality between packets and cells and the bonding (PAF) functionality. The Serial TPS-TC (TDM) includes an HDLC controller. The Multiplexer and M-pair block realizes the M-pair/Four-wire functionality The Integrated Device Controller (IDC) controls on one hand all internal blocks, on the other hand it provides host controller functionality in stand alone mode (see **Figure 3**).

The SOCRATES™-4e provides as system interface for packet and cell data the UTOPIA/POS-PHY interface or xMII interfaces. These interfaces are referred to as System Interface 1. For TDM data one interface is shared between the lines, it is referred to as System Interface 2. For controlling the SOCRATES™-4e a parallel microcontroller interface, a serial control interface (SCI) or an MDIO interface (only available if xMII is chosen) may be used. For connecting an external flash device an SPI interface is foreseen (can be used in the stand alone mode).

The following figure shows the functional block diagram of the SOCRATES™-4e.

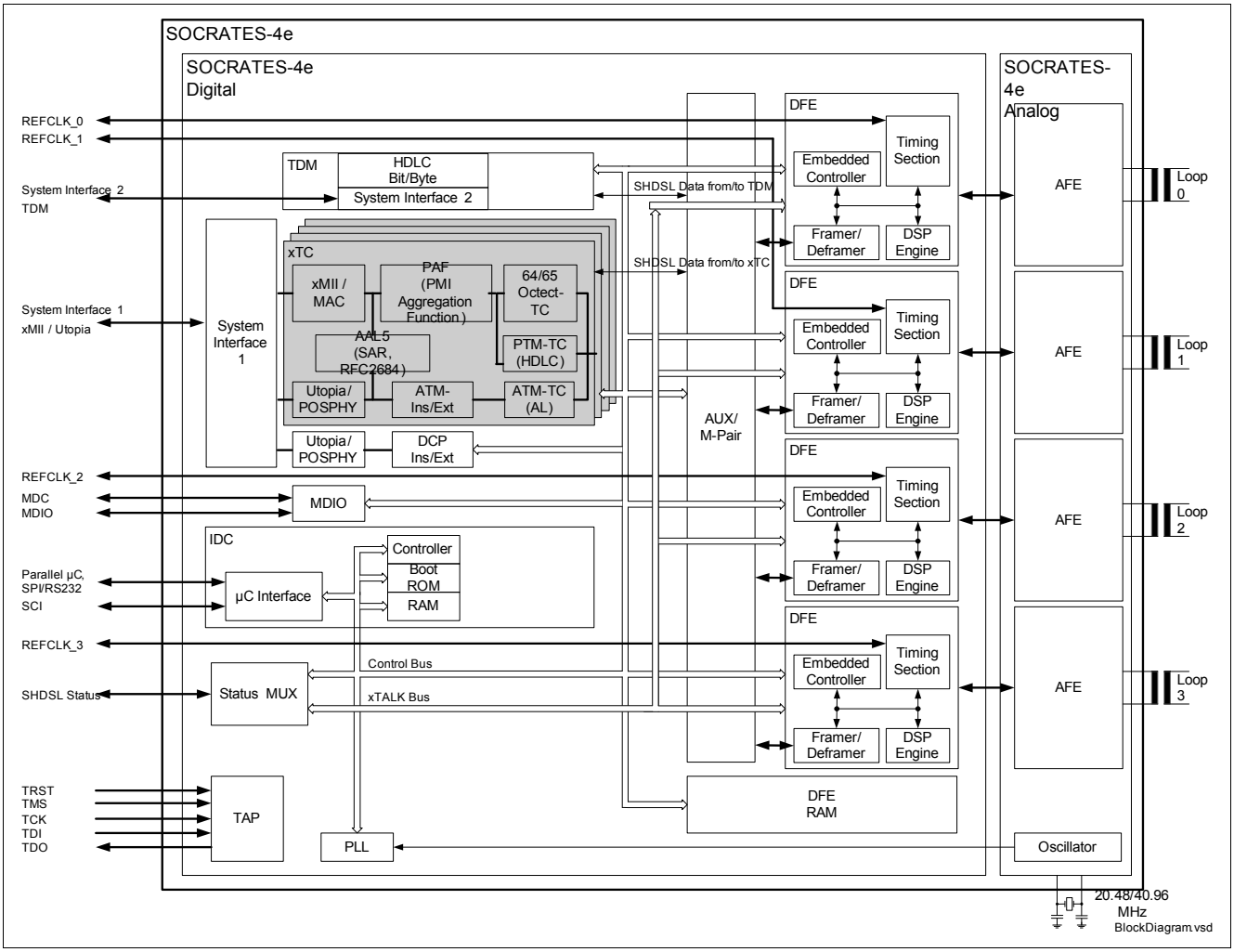


Figure 26 Block Diagram

4.2 Functional Description, Digital Part

The digital part consisting of the DSP including cross talk cancelling, the xTC including PAF, the Serial TPS-TC (TDM) including HDLC controller, the Multiplexer and M-pair block and the IDC is described in the following subclauses.

4.2.1 xTC Block Diagram

Figure 27 gives an overview of one xTC connected to System Interface 1. The additional connections of the PAF imply the connection to TCs of other xTCs to build an aggregation group.

Four xTCs are integrated in the SOCRATES™-4e (1 xTC per SHDSL channel).

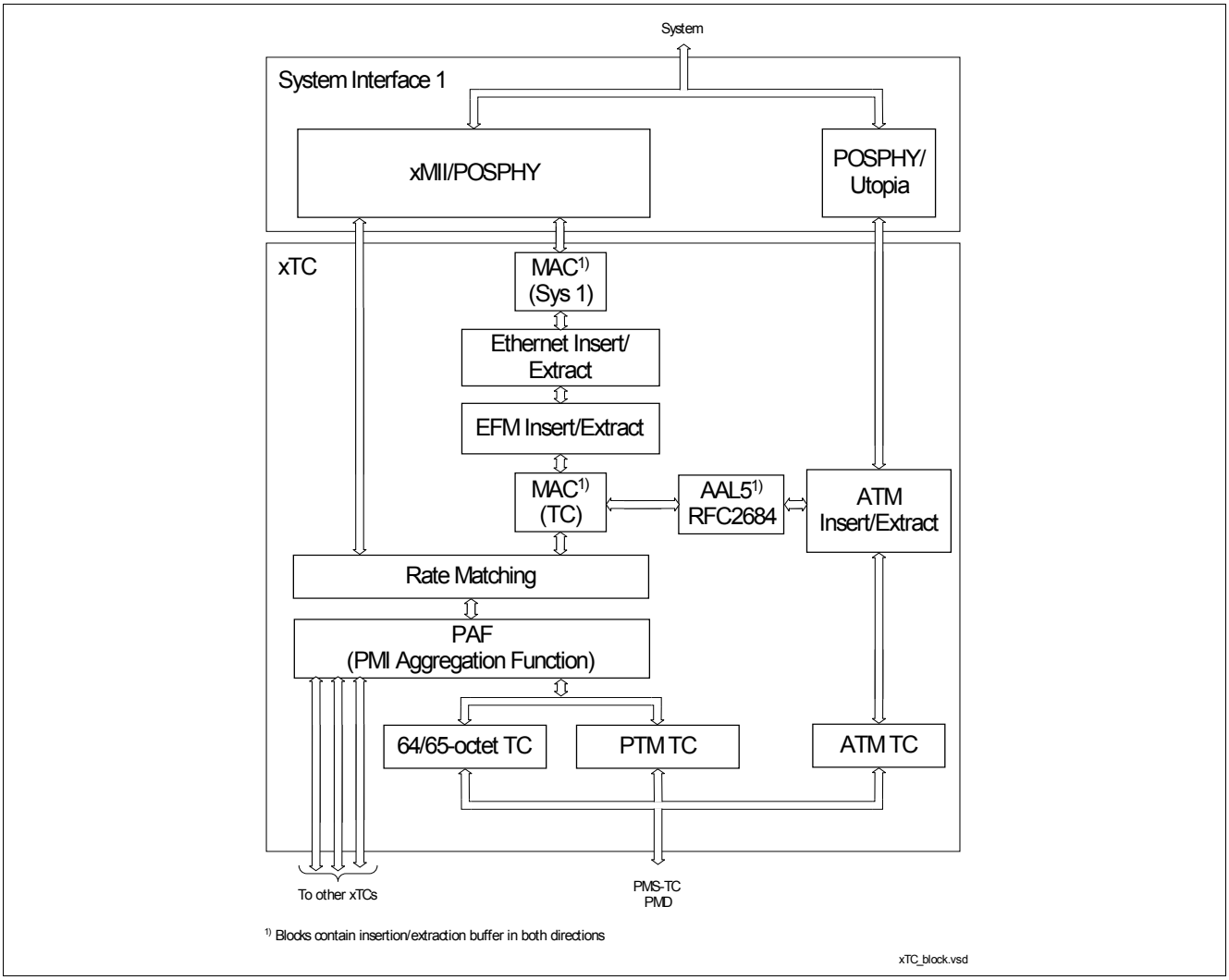


Figure 27 Block Diagram of the xTC

The following chapters show the functional blocks of the xTC which are used for supporting the different applications shown in section **Chapter 1.3.3**. Reference is always the block diagram given in **Figure 27**, used functional blocks for the application are highlighted.

4.2.2 CPE EFM PHY Mode Application

This chapters shows which functional blocks of the xTC are involved for the CPE EFM PHY Mode applications.

The following block diagram highlights the functional blocks of the xTC needed for realizing EFM PHY mode applications.

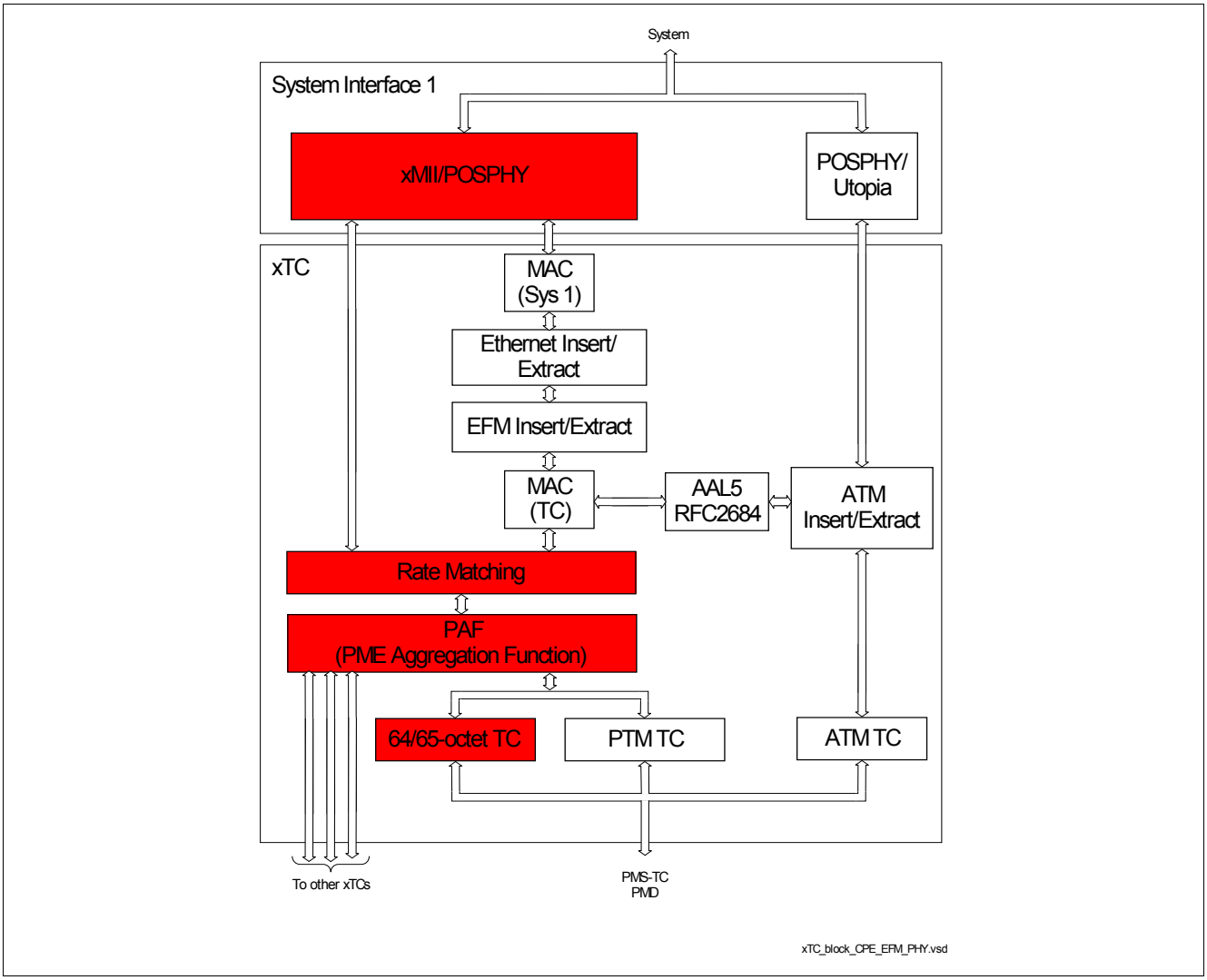


Figure 28 xTC Block Diagram, CPE EFM PHY Application

4.2.2.1 Rate Matching Block

The Rate Matching function is in compliance with IEEE 802.3-2004.

The block is directly connected to the xMII interface block. The MII interface is configured in the way to work with a MAC operating at 100 Mbit/s. The function of the block is the adjustment of the MAC's rate of data transmission to the transmission rate of the SHDSL link (2 BASE-TL). The MAC itself transmits data at a rate of 100 Mbit/s. Frames are buffered in the Rate Matching Block before transmitting them to the SHDSL links.

Rate Matching will be done using the CRS signal. This implies that the MII interface has to be operated in the so-called half duplex mode.

Before the MAC transmits a frame it checks the CRS signal. It will not transmit a frame as long as the CRS signal is asserted. In order to prevent the rate matching transmit buffer from overflow, the Rate Matching block keeps the CRS signal asserted until it has space to store a maximum length frame.

The Rate Matching transmit function strips the preamble and the SFD (start of frame delimiter) from the MAC frame and forwards this frame to the following blocks. In order to detect the beginning of a MAC frame properly the Rate Matching requires 1 byte of the preamble and the correct SFD.

In RX direction the Rate Matching function buffers a complete frame, prepends the Preamble (7 bytes) and SFD fields and sends this frame to the MAC. When sending the frame on the MII interface, the CRS signal will be asserted as well.

Some MACs, configured in half-duplex mode do not support the simultaneous transmission and reception of data. The SOCRATES™-4e supports this kind of MAC's by setting the variable tx_rx_simultaneously to FALSE. In this case sending the received frame to the MAC has always priority over receiving a frame from the MAC.

The Rate Matching block of the SOCRATES™-4e also supports MAC's operating with reduced MII interfaces (RMII, SS SMII). If this reduced MII interface infers a collision when TX_EN and CRS are both true, the variable crs_and_tx_en_infer_col has to be set to TRUE.

4.2.2.2 PME Aggregation Function (PAF) Block

The PAF functionality allows to bond several physical links to 1 logical link. Bonding itself takes place above the TC (in contrast M-pair/Four-wire mode where bonding takes place below the TC, at the physical layer). Once a PAF group is established over several links, links can be removed/added to the group seamlessly, group traffic is maintained and will be redirected to the remaining links (this behavior is in contrast to M-pair/Four-wire mode, where the remove of 1 link leads to break down of the entire group).

The PAF block allows the aggregation of up to 4 physical SHDSL (2BASE-TL) - referred to as PME's in IEEE 802.3-2004- links to 1 logical Ethernet link. The PAF block is located below the Rate Matching block and the TC layer. It has a single connection to the Rate Matching block and up to four connections to the TC of the SHDSL links forming the aggregation group.

This block can be bypassed in case aggregation is not required.

Feature List of CPE Application

- Fully compliant to IEEE 802.3-2004
- 1 PAF groups with up to 4 links per group (1 group of 4 links, 1 group of 3 links, 1 group of 2 links, 1 simple link)
- Maximum Differential Delay[1)] between any links of the group of 15000 bit times to be compensated (accdg. to IEEE 802.3-2004)
- All buffers integrated
- Speed Ratio of 4 between the fastest and slowest link of the group (accdg. to IEEE 802.3-2004)

The following pictures illustrates the PAF functionality.

1) Differential latency measures the variation in the time to transmit across different PMEs. To normalize the latency measurement for high and low speed links it is measured in bit times. A differential latency between two PMEs is defined as the number of bits N, that can be sent across the fast line, in the time it takes one maxFragmentSize fragment to be sent across the slow link. The value for differential latency for two identical links will be 4096 bit times because the definition includes the length of a maximum fragment size

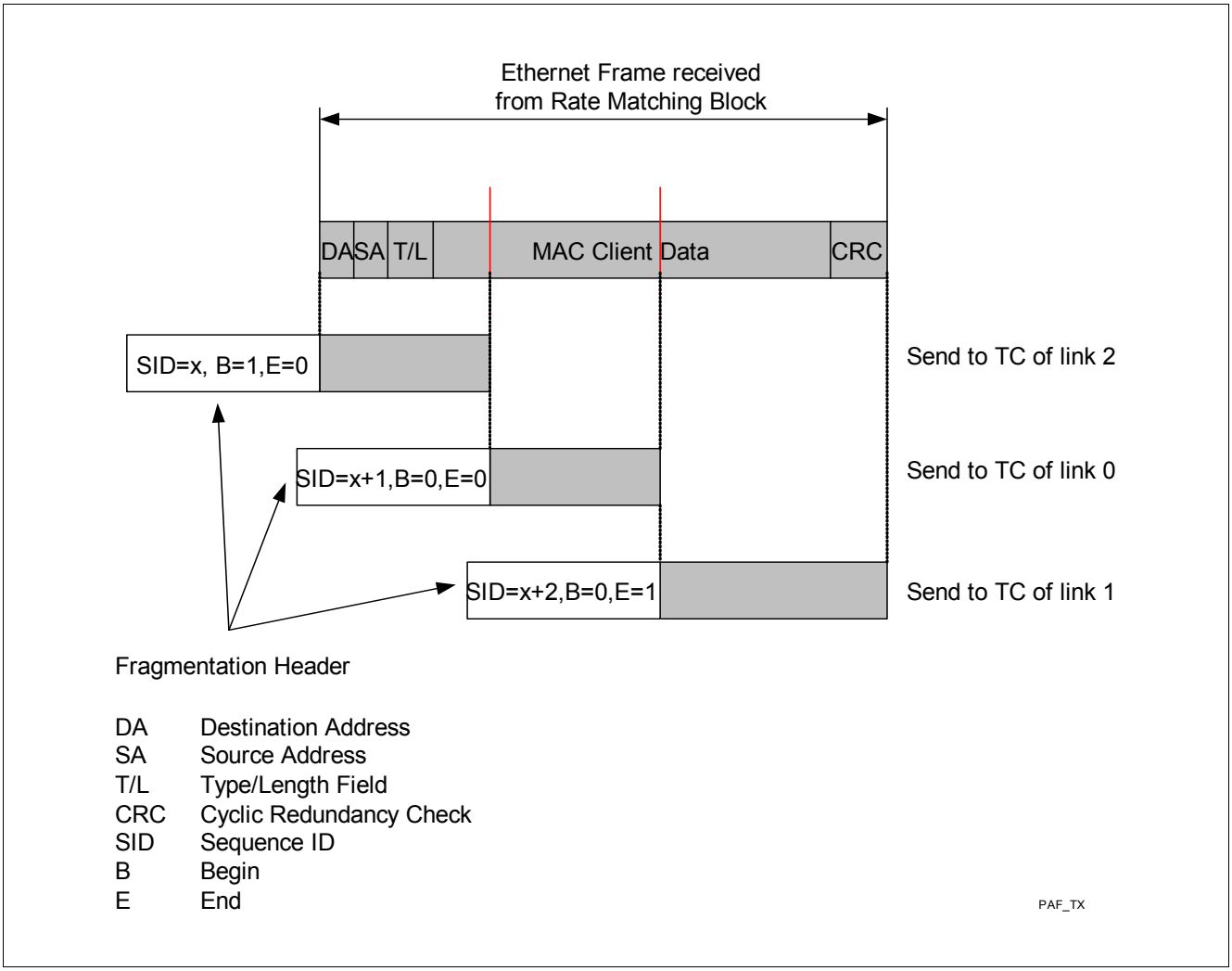


Figure 29 PAF Function

The PAF provides a fragmentation procedure at the transmitter and a reassembly procedure at the receiver.

The fragmentation procedure takes a MAC frame received from the Rate Matching Block and partitions it into one or more fragments (depending on the frame size of the entire MAC frame, as **Figure 29** MAC frame is partitioned into 3 fragments). A fragmentation header consisting out of a 14 bit sequence ID (SID), a begin flag and an end flag will be prepended before each fragment. These fragments will be sent to TC's belonging to the PAF group.

During fragmentation the following conditions have to be observed:

- Fragment Size within minFragmentSize and maxFragmentSize (specified to 64 byte and 512 byte according to IEEE 802.3-2004) not including PAF header
- Fragment Size, not including the PAF header, shall be a multiple of 4 octets except for the last fragment of a MAC frame

The SOCRATES™-4e does the fragmentation into fragments of equal sizes. The default fragment size applies to all links, is 256 byte and configurable. This default size will not be applied for the last 2 fragments belonging to a MAC frame. For every fragment the SID is incremented by 1 and the B and E flags will be set appropriately. Note that the SID rolls over after it reaches the maximum value- Distribution of the fragments to the TC sublayer of the link belonging to a PAF group will be done in a round robin way.

The reassembly procedure (PAF receive function) requires queues per SHDSL link and a per MAC buffer for fragment assembly. The size of each link queue is determined by the maximum differential delay and the size of the MAC buffer by the maximum supported frame size (configurable). The receive algorithm works based on an

expected fragment sequence numbers (SID) and next sequence number. Queues where the SHDSL link is UP and the TC is synchronized will be considered in the receive algorithm. This algorithm has the following steps:

- Step1: Determine the expectedFragmentSequenceNumber (after reset, this value will be set to the smallest FragmentSequenceNumber)
- Step2: Determine the nextFragmentSequenceNumber (nextFragmentSequenceNumber will be set to the smallest FragmentSequenceNumber available at the top of each active queue)
- Step3: IF expectedFragmentSequenceNumber = nextFragmentSequenceNumber, pull the respective fragment into reassembly buffer, if this fragment is an end-of-packet fragment pass the entire MAC frame to the Rate Matching Block
- Step4: Increment (modulo 2^{14}) expectedFragmentSequenceNumber
- Step5: Continue with Step2

Errors during Reassembly

In case of errors during the reassembly process, they will be classified into 3 groups:

- Errors during fragment reception
- Errors in fragment sequencing
- Errors in packet reassembly

The following errors are classified as errors during frame/fragment reception: Rx_Err asserted during reception of the fragment, the fragment is too small (less than minFragmentSize), the fragment is too large (more than maxFragmentSize), the fragment would cause the per PME queue receive buffer to overflow.

The following errors are classified as errors in fragment sequencing: nextFragmentSequenceNumber is outside the range (counted as bad fragment received), all PMA queue buffers are non empty and nextFragmentSequenceNumber is greater than expectedFragmentSequenceNumber (counted as lost fragment), any PMA queue buffer non empty for maxDifferentialDelay bit times and no fragment is transferred (counted as lost fragment as well).

The following errors are classified as errors in packet reassembly: a fragment is received with StartofPacket bit deasserted while the packet reassembly function was between frames (counted as PAF lostStart), a fragment is received with StartofPacket bit asserted while the packet reassembly function was mid-frame (counted as PAF lostEnd), a fragment is received while the packet assembly function was mid-frame and would cause the frame size to exceed the maximum allowable frame size (counted as PAF lostEnd).

In all cases a frame will be sent over the MII interface to the MAC. If the packet reassembly function was mid-frame, the first part of the frame is transferred across the MII interface, assert the RX_ER signal on the MII and PMA queue buffers will be flushed until the next Start of Packet. If the packet reassembly function however was between frames, the RX_ER will be asserted on the MII interface and a garbage frame (as defined in IEEE) will be sent.

Additionally for all the different error conditions dedicated counters will be maintained.

PAF Configuration (via g.handshake)

PAF will be configured during the g.hs phase. New g.hs identification code points were defined for the support of PAF function (see Ammendment2 (06/2004) of G.994.1 (2003)). PAF is part of the bonding (ID field, SPar(1) codings, octet 3) and referred to as Ethernet bonding within g.hs (ID field, NPar(2) codings. The PAF functionality itself will negotiated using the ID field SPar(2)/NPar(3) codepoints of the Ethernet/TDIM bonding tree.

The configuration is spit into 2 phases, an aggregation discovery phase and an aggregation phase. During the aggregation discovery phase all potential links which might be aggregated to 1 PAF group will be identified. During the aggregation phase, physical links (PME entities) will be aggregated to a common PAF group. This entire configuration phase is driven from the CO device, certain adjustments need to be done on the CPE side before the configuration phase starts:

General PAF support will be identified by PAF available bit (see PCS register 10P/2B capability register). The SOCRATES™-4e and SOCRATES™-2e both support PAF in CPE EFM applications.

Afterwards the number of supported links per PAF group might be limited (see PCS register 10P/2B PME available register). Per default and after reset the SOCRATES™-4e can aggregate all 4 links (PME, PME available (3.62) = 0000_H, PME available (3.63) = $000F_H$), the SOCRATES™-2e can aggregate 2 links (PME, PME available (3.62) = 0000_H, PME available (3.63) = 0003_H). This capacity can be limited by accessing this PME available register, in both application PAF will be controlled in PCS 0. Note that any limitation deviating from the reset value need to be configured before g.hs starts. First of all the number of supported PAF groups needs to be adjusted.

The discovery phase itself takes place on every link (PME). The CO is accessing the so called remote_discovery_register which is only defined for CPE application. The remote_discovery_register is 48 bit wide and only implemented once (per PCS instance). Access to this remote_discovery_register from the different links (PME) will be controlled using the PME available registers. Only PMEs having the appropriate PME available bit set can access the remote_discovery register. The following figure illustrates this behavior, in the given example the PAF ability of the SOCRATES™-4e is limited to links 0,1,2.

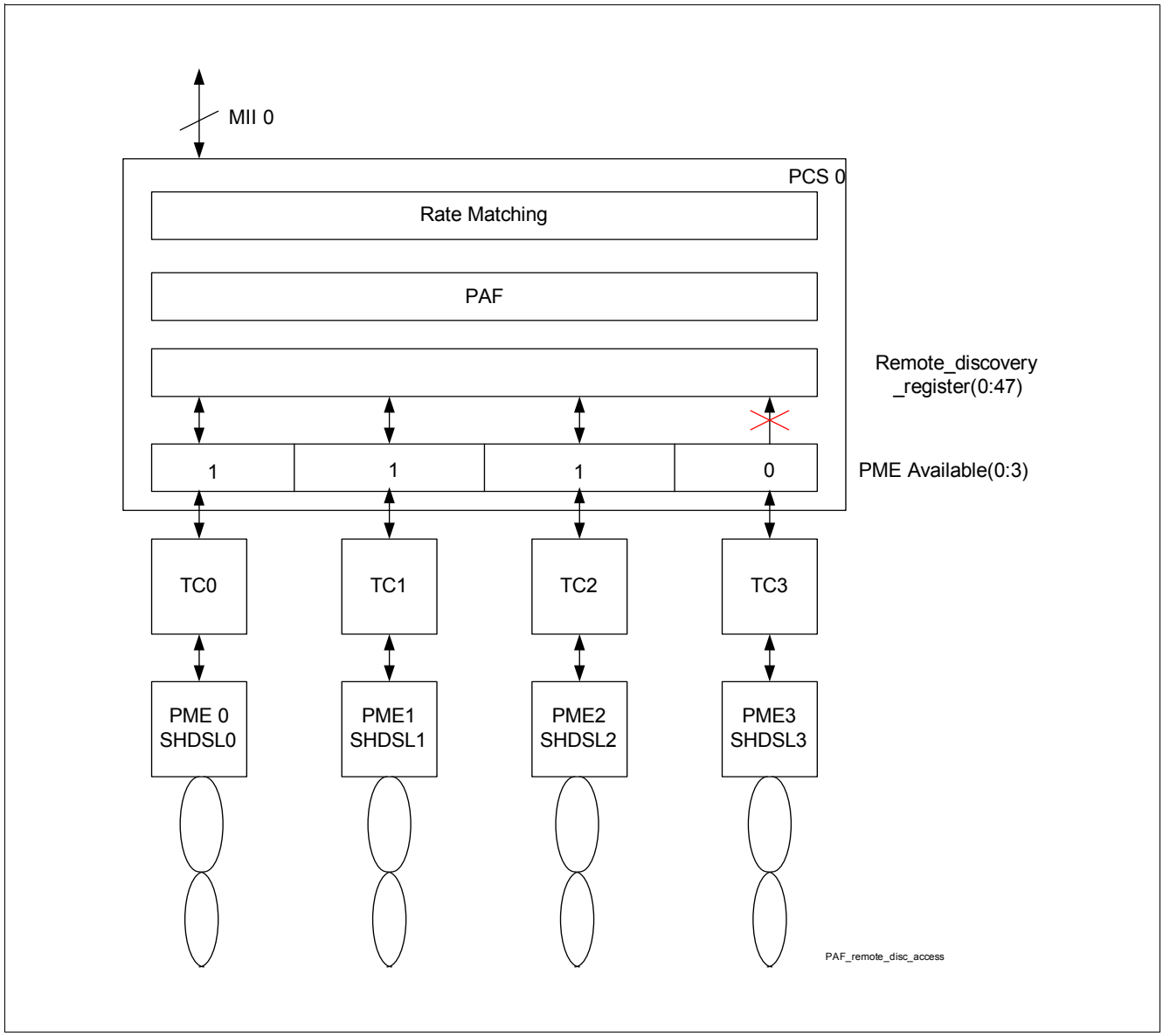


Figure 30 Access to Remote_discovery_register

The reset value of the remote_discovery_register is 0. For accessing the remote_discovery_register (if enabled by appropriate bit in PME Available register) 2 atomic operations are defined:

- Set if Clear
  If the remote_discovery_register is currently clear (no bit set), the content of remote_write_data (data sent from the CO as part of the Set if Clear command) is placed into the remote_discovery_register. The new contents of the remote_discovery_register is placed on remote_read_data (which is sent back to the CO device). Note that these 2 operations (Write and Read) are atomic. If the remote_discovery_register is currently not clear, no new data will be placed into the remote_discovery_register, but the old contents of remote_discovery_register will be placed on remote_read_data
- Clear if Same
  If the contents of remote_write_data match that of the remote_discovery_register the remote_discovery_register and the PME_aggregate_register as well will be cleared and the new contents of the remote_discovery_register will be placed on remote_read_data. If the content of the remote_write_data does NOT match that of remote_discovery_register, the contents of remote_discovery_register remain unchanged and this content will be put on remote_read_data. Note that these 2 operations are atomic.

Note that the Amendment 2 of G.994.1 contains only 1 code point for these 2 operations. If the code point ’Clear if same’ is set, it is a ’Clear if Same’ operation, else it is a ’Set if Clear’ operation.

In order to find out potential links for aggregation the CO might start a Set if Clear operation on all links belonging to the CO, where the content of remote_write_data has to be unique on each line (i.e. MAC address). Based on the received data (remote_read_data) the CO can determine which links are available for aggregation. The following figure illustrates this behavior.

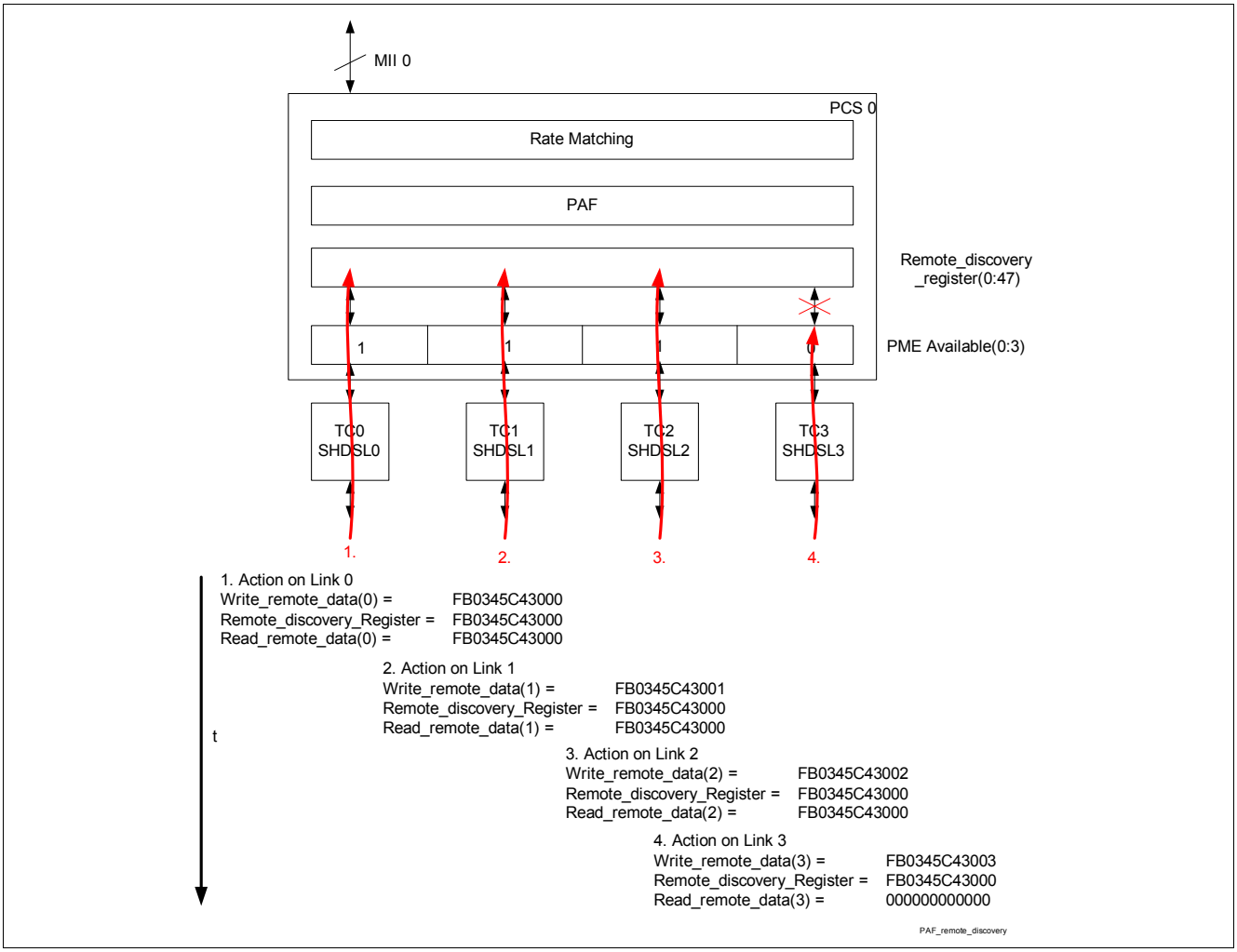


Figure 31 Remote Discovery Procedure

Based on the read_remote_data values from different links the CO can determine which links can be aggregated. In the example given in **Figure 31** links 0, 1 and 2 can be aggregated.

Aggregation itself will be programed using the PME_aggregate_register. This register is available at both sides. It will be written locally at the CO side and using g.hs at the CPE side. Amendment 2 of G.994.1 defines under the PME aggregation codepoint a 32-bit wide code point field used for PME aggregation (PME_aggregate_register bits). The procedure for accessing the remote PME_aggregate_register is similar to remote discovery process. Bit 0 of the g.hs field ‘PME_aggregate_register’ will be set on all links belonging to a group and be sent to the CPE. At the CPE side bit 0 is written to the PME_aggregation_register in the bit location corresponding to the PMA/PMD from which this request was received. The following figure illustrates this behavior.

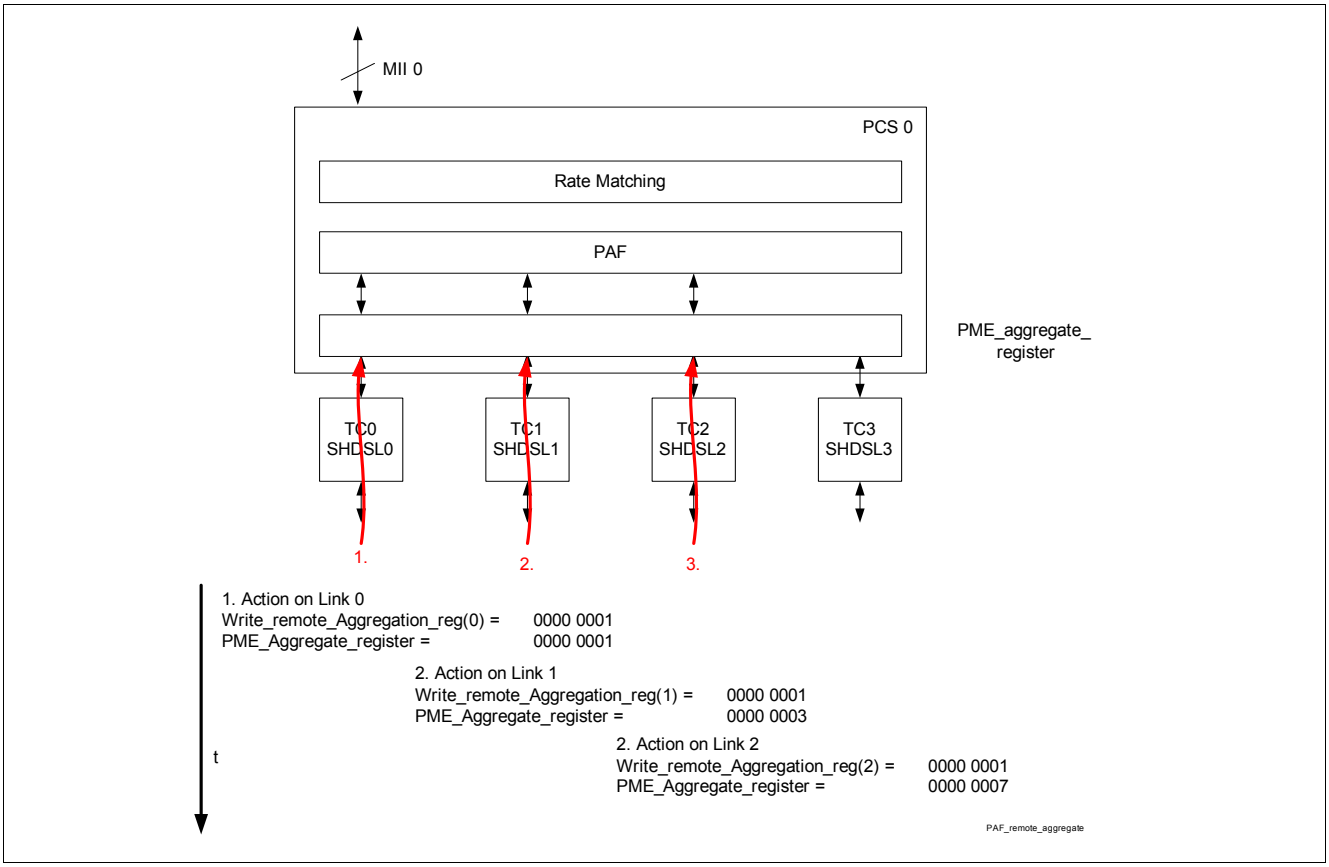


Figure 32 PME Aggregation Procedure

Note that the PME_Aggregate_register has to be programmed identically on both the CO and the CPE side in order to guarantee a correct bonding function. Afterwards the links will be activated and the bonding group is configured and set up.

PAF Configuration (via BACP Protocol)

Negotiations of PAF groups via g.handshake implies certain limitations (i.e. seamless assignment of links to different groups requires a new g.handshake sequence). These limitations were addressed by the Bonding Aggregation Control Protocol (BACP) and is defined in G.998.2 Amendment 2.

Support of BACP will be negotiated via a newly defined handshake code point. PAF groups will be configured either via g.handshake or the BACP protocol, a combination of both methods is not possible.

The BACP protocol uses Ethernet Frames with a unique identification (type field = “slow protocol” Ethernet frames). These frames will be exchanged between the STU-C and STU-R once the SHDSL links are up and running and internally processed by the SOCRATES™-4e/SOCRATES™-2e. The BACP protocol defines procedures for setting up and turning down PAF groups as well as a seamless addition/removal of links to/from a group. The protocol furthermore provides means to allocate links to different groups (over time) without a link down, one link however can only belong to one PAF group at a certain time. PAF groups will always be started with a group of 1 link.

4.2.2.3 64/65 octet TC Block

This blocks works below the gamma interface. Since the PAF functionality is optional either fragments or entire Ethernet frames may be passed by the TC block across the gamma interface. In this section, the term ’fragment’ will be used in order to describe either fragments or Ethernet frames.

This block provides a full transparent transfer of data fragments between the gamma interface at the CO side and the gamma interface at the CPE side. In transmit direction, the TC receives fragments from the PAF via the gamma interface. A CRC is calculated on the data and appended to the fragment. The TC then performs the 64/65-octet encapsulation and sends the resulting codewords to the PMA. In receive direction the TC receives codewords from the PMA. Then it recovers the transported TC fragments and checks the CRC. The extracted fragments will be submitted across the gamma interface to the PAF.

The TC also generates special Idle Codewords in case the PAF does not provide fragments

TC Encapsulation and Coding

The TC coding function generates codewords with a fixed length of 65 octets. A codeword consists out of a Sync Octet and one of the following combinations:

- All data: all of the octets in the codeword belong to the same TC fragment
- End of frame (go to idle): up to 63 octets in the codeword belong to the same TC fragment, the rest of the codeword consists of Idle octets
- End of frame (Start of new frame): up to 62 octets in the codeword belong to the same TC fragment, a number of Idle octets and a single Start of Frame octet precede the first data octets of the next TC fragment
- Idle: all of the octets in the codeword are Idle octets
- Idle (start new frame): a number of Idle octets and a single Start of Frame octet precede up to 63 data octets of the next frame
- Out-of-sync idle: all of the octets in the codeword are idle octets and the 64/65-octet receive state machine is out_of_sync

In case of the 2 end of frame codewords a special octet (Ck) follows the Sync octet. The value of Ck tells how many valid data octets follow the Ck octet and belong to the same TC fragment. The following table shows all the special TC control characters:

Table 26 TC Control Characters Values (accdg. to IEEE 802.3-2004)

Character	Value (in hex)	Description
Sync octet (all data)	0F	Value of Sync octet in case of all data codewords
Sync octet (else)	F0	Value of Sync octet in all other cases
Idle	00	Idle character (used for insertion of Idle octet)
Ck	k+10 (MSB even parity)	Ck character defines the number of remaining data octets belonging to the TC fragment within this codeword, k=0+63 (number of data octets)
Y	D1	Out of Sync Marker, this octet will be used in the first position after the Sync octet in order to mark an out-of-sync codeword
S	50	Start of Frame octet
Z	00	Idle Octet

Based on this table the following code violations can be detected (signal TC_coding_error asserted):

- Incorrect octet received when a Sync Octet is expected
- Outside a fragment, the received octet following a valid $F0_{16}$ is not a Z, Y, S
- Inside a fragment the received octet following a valid $F0_{16}$ is not a valid value of Ck
- Z or S is expected, and a value different from Z and S is received

TC CRC Function

The TC-CRC is generated for the entire payload fragment including any attached header (from PAF) including the Ethernet CRC. The TC CRC is appended to the data stream at the end of the fragment in transmit direction and

the TC-CRC is checked against the last 4 octets of the fragment in receive direction. If the received TC-CRC is incorrect then Rx_Err is asserted to signal that the fragment is errored. The CRC function is defined by the following polynomial:

$x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+1$

If in transmit direction the TX_Err signal is asserted during the transmission of a fragment across the gamma interface, the last octet of the TC-CRC is ones-complemented.

Alternatively a 16 bit CRC can be generated and appended.

Bit ordering

Payload data will be sent to the PMA/PMD LSB first, the CRC will be MSB first

Sync Detection

Sync Detection will be applied in receive direction. The sync detection serves 2 purposes. First the synchronization is acquired from the incoming data stream, the sync detection controls the initial acquisition and maintenance of the synchronization. Secondly, based on correct sync detection the receive control state machine can extract framing information from the receive data stream and remove the sync octets and the CRC codes.

The sync detection works on sync octets (both sync octet $0F_{16}$ and $F0_{16}$ - see **Table 26**, referred to as Sync in **Figure 33**) and is shown in the following figure (accdg. to IEEE 802.3-2004):

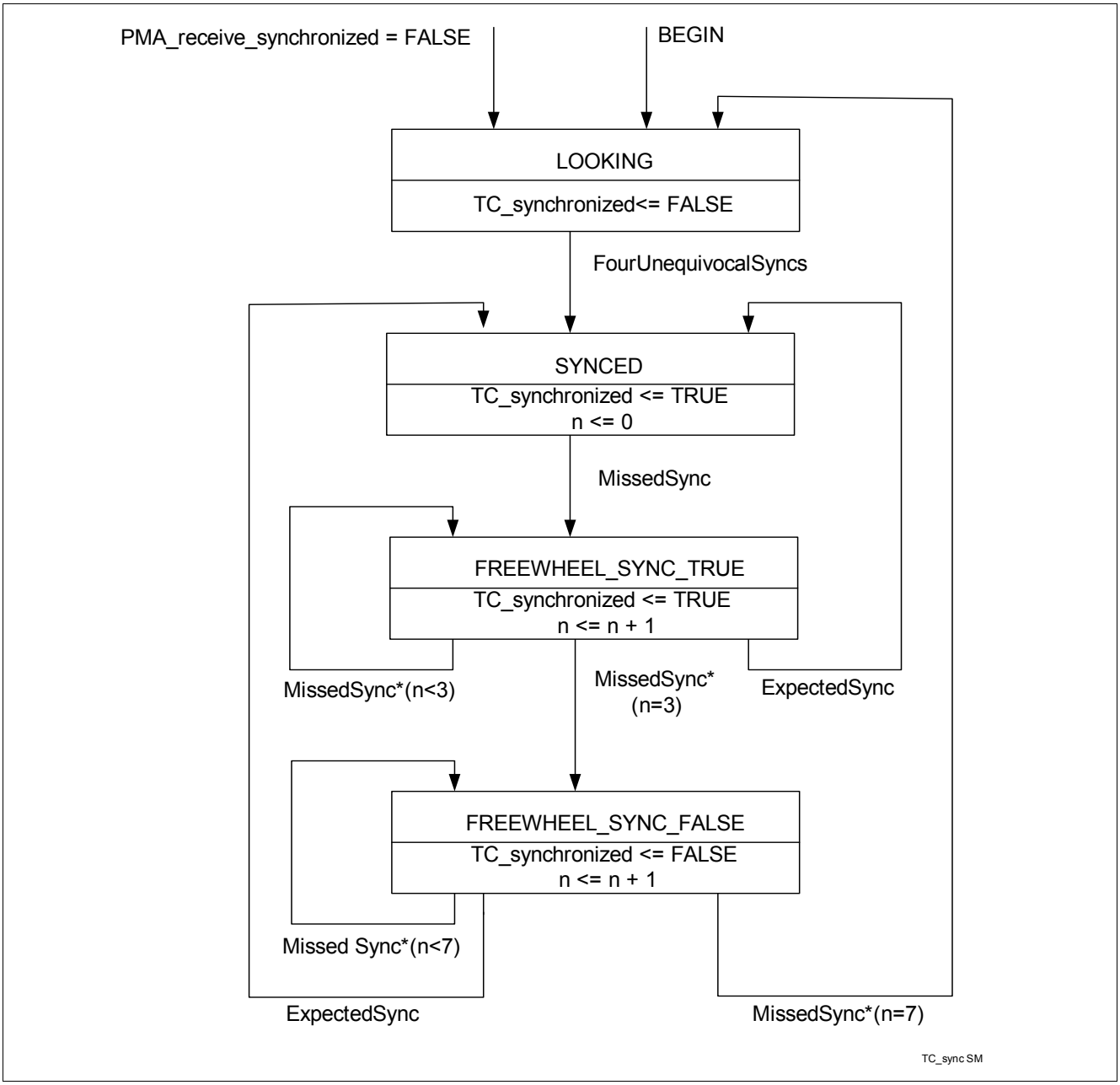


Figure 33 TC Synchronization Detect State Machine

PMA_receive_synchronized refers to the synchronization of the DSL framer (LOSW = ’0’). As long as TC_synchronized is TRUE, the TC sends data to the other blocks.

4.2.3 CPE EFM MAC Mode Application

This chapter shows and describes the functional blocks of the xTC involved for realizing the CPE MAC Mode. The following figure highlights these blocks. Note that this section only contains the description of new blocks which are not covered in **Chapter 4.2.2**.

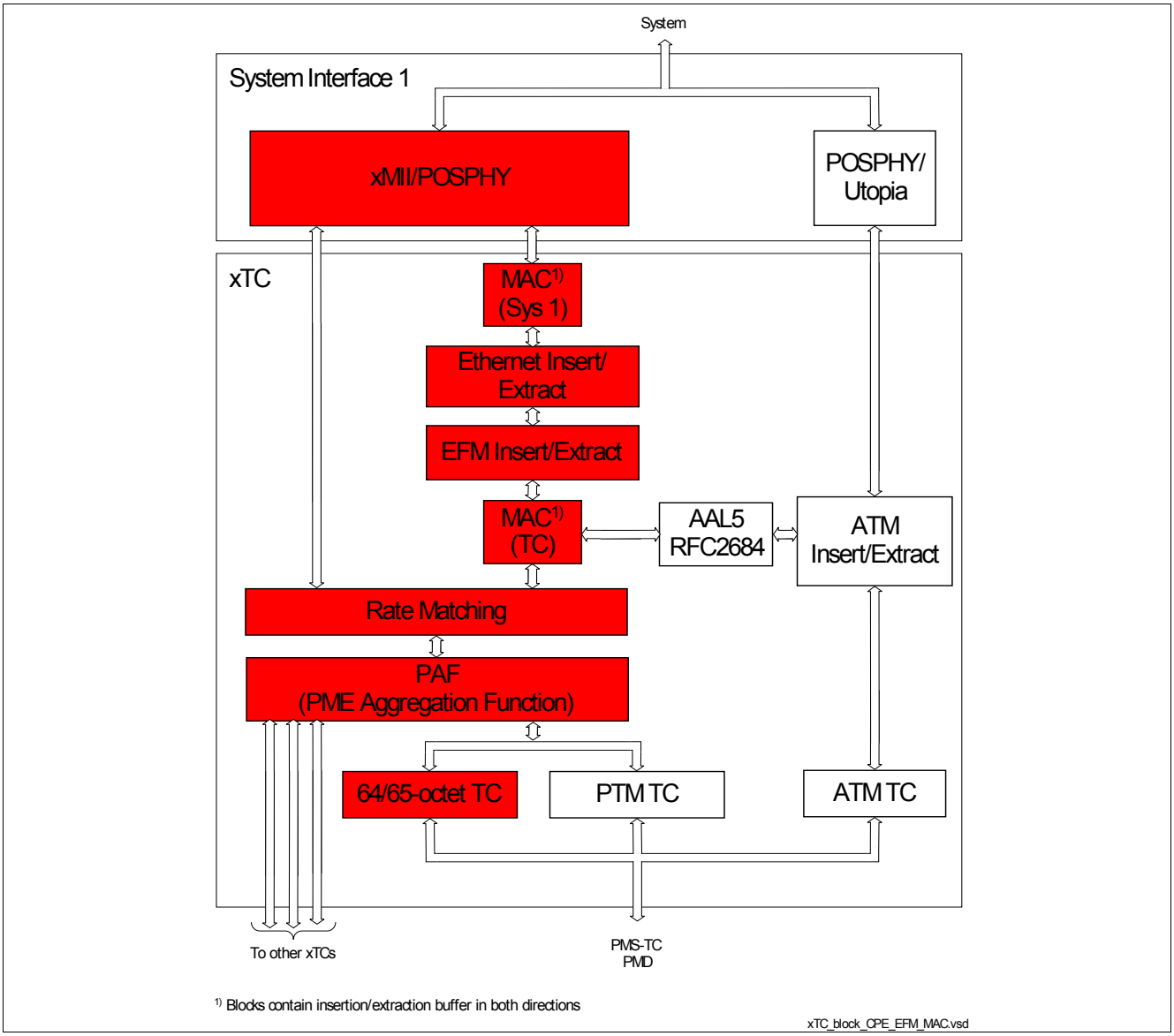


Figure 34 xTC Block Diagram, CPE EFM MAC Mode

4.2.3.1 MAC Block

Both MAC's (Sys 1;TC) are in compliance with IEEE 802.3-2004. According to this standard the functionality of the MAC block can be spit in 2 tasks:

- Data Encapsulation
- Media Access Management

Data Encapsulation

Data encapsulation itself consists of 3 parts, framing, addressing and error detection. Mainly all frames will be transparently passed through the SOCRATES™-4e device. Due to this approach the framing and addressing function are nil. For error detection the MAC block supervises the frame length (minimum frame size, maximum frame size) and does a FCS generation/comparison. Frames which do not pass these checks will be dropped.

Media Access Management

The MAC at the TC side is directly connected to the Rate Matching block, so this MAC works according to IEEE 802.3-2004 in half duplex mode. Towards the system side, 100 Mbit/s Ethernet PHY's can be directly connected, both half and full duplex modes are supported.

In case of half duplex mode the CSMA/CD protocol is supported, in case of full duplex mode no special protocol needs to be applied (assumption: physical media is able to transmit and receive simultaneously, exactly 2 station connected to the media, transmission is full duplex).

This MAC (Sys 1) is for direct connection of an Ethernet PHY and covers the layer 2 MAC functionality. Incoming packets with incorrect CRC and wrong sizes are dropped.

In order to support various MIB the following table lists the set of counter which are implemented per MAC on each side (TC and system). Transmit direction always refers to the direction from MAC to the media, receive direction to the direction media to MAC.

Table 27 MAC MIB Counter

Counter Name	Half/Full Duplex	MIB, Element Name	Comment
Counter Transmit Direction			
Transmit Unicast Packet Count	H/F	RFC 2863, ifOutUcastPackets	Number of valid unicast frames without MAC control and Slow Protocol frames
Transmit Multicast Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifOutMulticastPkts	Number of valid multicast frames without MAC control and Slow Protocol frames Note: a broadcast frame is not a multicast frame
Transmit Broadcast Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifOutBroadcastPkts	All frames with DA = 1 (all bits) without MAC control and Slow Protocol frames
Transmitted Frames	H/F	IEEE 802.3-2004, Clause 30	
Single Collision Frames	H	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsSingleCollisionFrames	Frames that experienced a single collision and were transmitted afterwards
Multiple Collision Frames	H	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsMultipleCollisionFrames	Frames that experienced multiple collision (>1) and were transmitted afterwards
Excessive Collision Count FramesAbortedDueToXSColls	H	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsExcessiveCollisions	Frames which were not transmitted due to excessive (>16) collisions
Transmit Late Collisions	H	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsLateCollisions	Frames which experienced a late collision (after 512 bit) Note: this collision will also be counted as normal collision

Table 27 MAC MIB Counter (cont'd)

Counter Name	Half/Full Duplex	MIB, Element Name	Comment
Deferred Transmission FramesWithDeferedXmissions	H	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsDeferredTransmissions	Frames where transmission was deferred due to busy media - deferring ON (only during initial transmission relevant), Frames which experienced a collision will not be counted
Transmit MAC Control Frames	H/F	IEEE 802.3-2004, Clause 30	Sum of all MAC Control Frames (accdg. to IEEE 802.3-2004, Annex 31A)
Transmit Pause Frames	F	IEEE 802.3-2004, Clause 30 RFC 2665, dot3OutPauseFrames	
Transmit Slow Protocol Frames	H/F		Sum of all Slow Protocol Frames (accdg. to IEEE 802.3-2004, Annex 43B)
Transmit OAM Frames	H/F	IEEE 802.3-2004, Clause 30	Slow Protocol Frames, OAM Subtype
Transmit Byte count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifOutOctets	Count of all octets in valid sent frames (not including SFD and preamble)
Counter Receive Direction			
Receive Byte Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInOctets; RFC 2819, etherStatsOctets	Count of all octet in valid frames (not including SFD and preamble)
Receive Unicast Packet Count	H/F	RFC 2863, ifInUcastPkts	Count of valid unicast frames (without MAC control and Slow Protocol frames)
Receive Multicast Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInMulticastPkts; RFC 2819, etherStatsMulticastPkts	Count of valid multicast frames, without MAC control and Slow Protocoll frames
Receive Broadcast Frames	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInBroadcastPkts; RFC 2819, etherStatsBroadcastPkts	Count of valid broadcast frames without MAC control and Slow Protocoll frames Note: A broadcast frame is not a multicast frame.
Receive Frames OK	H/F	IEEE 802.3-2004, Clause 30 RFC 2819, etherStatsPkts	Correct Received Frames
Receive CRC Error Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsFCSErrors; RFC 2819, etherStatsCRCAlignErrors	Frames with CRC errors, address don't care, length within boundaries and length integer number of bytes
Receive Alignment Error	H/F	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsAlignmentErrors	Frames with alignment errors, address don't care, length no integer number of octets and FCS checks fails

Table 27 MAC MIB Counter (cont'd)

Counter Name	Half/Full Duplex	MIB, Element Name	Comment
Receive Dropped Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInDiscards; RFC 2819, etherStatsDropEvents; RFC 2665, dot3StatsInternalMacReceiveErrors	Correct frames which were internally dropped (i.e. in Pause situation), does not apply to MAC/Slow protocol frames with unsupported opcodes
Receive Undersized Good Packet	H/F	RFC 2819, etherStatsUndersizePkts	Length < 64 byte, no error
Receive Undersized Error Packet	H/F	RFC 2819, etherStatsFragments	Length < 64 byte, error
Receive Size 64 Packet Count	H/F	RFC 2819, etherStatsPkts64Octets	
Receive Size 65-127 Packet Count	H/F	RFC 2819, etherStatsPkts65to127Octets	
Receive Size 128-255 Packet Count	H/F	RFC 2819, etherStatsPkts128to255Octets	
Receive Size 256-511 PacketCount	H/F	RFC 2819, etherStatsPkts256to511Octets	
Receive Size 512-1023 PacketCount	H/F	RFC 2819, etherStatsPkts512to1023Octets	
Receive Size 1024-1518 PacketCount	H/F	RFC 2819, etherStatsPkts1024to1518Octets	
Receive Oversize GoodPacketCounter	H/F	RFC 2819, etherStatsOversizePkts	Length > max length, no errors
Receive Oversize Error Packet Count	H/F	IEEE 802.3-2004, Clause 30 RFC 2665, dot3StatsFrameTooLongs	Length > max length, errors
Receive MAC Control Frames	H/F	IEEE 802.3-2004, Clause 30	Type field = 8808, DA either 01 80 C2 00 00 01 or own MAC address
Receive Pause Frames	F	IEEE 802.3-2004, Clause 30 RFC 2665, dot3InPauseFrames	Type field = 8808, DA either 01 80 C2 00 00 01 or own MAC address, OPC = 00 01
Receive MAC Control Frames Unknown Opcode	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInUnknownProtos	Type field = 8808, DA either 01 80 C2 00 00 01 or own MAC address, OPC /= 00 01
Receive Slow Protocol Frames	H/F		Type field = 8809, DA either 01 80 C2 00 00 02 or own MAC address
Receive OAM Frames	H/F	IEEE 802.3-2004, Clause 30	Type field = 8809, DA either 01 80 C2 00 00 02 or own MAC address, Subtype = 3
Receive Slow Protocol Frames (unknown subtype)	H/F	IEEE 802.3-2004, Clause 30 RFC 2863, ifInUnknownProtos	Type field = 8809, DA either 01 80 C2 00 00 02 or own MAC address, Subtype /= 3

Rate Adaption

Rate adaption of the different speeds between the LAN segment and the SHDSL segment will be realized between the 2 MAC's. 2 modes are supported, the working mode of the LAN segment (full duplex, half duplex) dictates the rate matching algorithm. Rate matching itself will be triggered based on filling levels of memories in the Rate Matching block (see **Chapter 4.2.2.1**).

If the LAN segment connected to System Interface 1 works in half duplex mode, rate matching will be done forcing a collision. As long as the transmit buffer of the Rate Matching block is filled that it cannot accept an Ethernet frame with maximum size, every frame on the LAN segment sent to the MAC (Sys1) will be forced to a collision. This is done by sending dummy data as soon as the reception of data is detected.

If the LAN segment connected to System Interface 1 works in full duplex mode, rate matching will be done sending PAUSE frames (as standardized according IEEE 802.3-2004, Annex 31). Sending these PAUSE frame is done with highest priority. The destination address of the PAUSE frame is the globally reserved multicast address as defined in the above mentioned standard, the type field is set to 8808_H, the opcode set to $00\text{-}01_H$. The source address of these frames is programmable and has to be set before starting operation. In order to address the latency caused by the PAUSE frame, buffers capable of storing more than 1 Ethernet frame are foreseen. Based on the filling level of these buffers PAUSE frame generation will be turned on and turned off, whereas different watermarks are available (1 for turning PAUSE frame generation on, 1 for PAUSE frame generation off).

The SOCRATES™-4e also supports the reception and processing of PAUSE frames. In order to clearly identify PAUSE frame, the destination address of these frames has to be set either to the globally reserved multicast address or to the own MAC address. In any case, the type field has to be set to 8808_H and the opcode set to $00\text{-}01_H$. PAUSE frame operation itself will be supported according the IEEE 802.3-2004, Annex 31B).

The MAC block also contains a management block for controlling a Ethernet PHY and gathering status information from the PHY. Frame format and management protocol are supported as standardized in IEEE 802.3-2004, Clause 22, the MAC block can access all registers of the MII management register set.

4.2.3.2 Extract/Insert Block

The extract/insert block may be used for extraction/insertion of Ethernet frames. Extracted frames will be either processed by the internal device controller (IDC, i.e. Ethernet OAM frames) or by the external device controller. Frames to be inserted are either generated by the IDC or by the external controller.

General Extraction

In total 2 extraction filter per direction are available (2 filter for device TX direction - System Interface 1 -> SHDSL interface; 2 filter for device RX direction SHDSL interface -> System Interface 1), so in total 4 extraction filter are available per channel.

Each filter is defined by 4 features:

- Search pattern
- Mask field
- Offset of Search Pattern
- Filter Characteristics

Each of these features can be programmed independently for each filter.

The search pattern is 48 bit wide and can be freely programmed. The mask allows a bitwise mask of the search pattern and is therefore also 48 bit wide (respective bit set to '1', bit will not be compared). The offset (given in bytes) defines the offset with respect to the beginning of the frame where the search mask shall be applied. 2 filter characteristics can be applied, extract and no forward (frame will be extracted and not forwarded within the data stream) or extract and forward (frame will be extracted and also forwarded within the data stream). A typical application for the second filter characteristic is a broadcast frame. The following figure illustrates the meaning of the 3 different filter features Search Pattern, Mask Field and Offset:

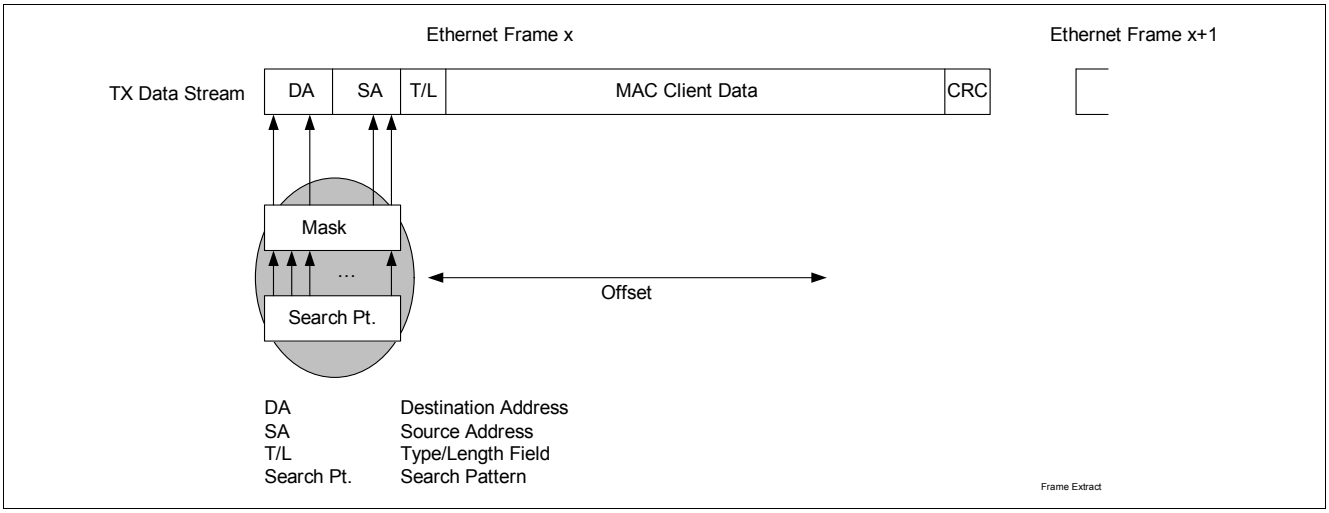


Figure 35 Frame Extract

Extracted data will be sent to an internal buffer, the IDC access this buffer and starts further processing of the frame. 1 such buffer is available per direction. If the buffer is not served by the IDC and a new frame fits the matching patterns, this new frame will be dropped. A dedicated counter counts all dropped frames.

General Insertion

The IDC can insert frames. Therefore it writes the frame to the insertion buffer and specify the channel number and the direction where the frame should be inserted. Afterwards this frame will be inserted with highest priority.

Ethernet OAM

The SOCRATES™-4e supports Ethernet OAM (accdg. to IEEE 802.3-2004, Clause 57) over the SHDSL segment. This module contains 2 blocks, the extraction/insertion block and the OAM block. The extraction part of the extraction/insertion block contains a filter, which is 9 byte wide. The following table contains the default setting of this filter:

Table 28 Default Filter Settings for OAM Support

Byte	Value [hex]	Meaning
1-6	01-80-C2-00-00-02	Reserved Address for MAC Control Frames
7-12	Don’t care	Source address not relevant
13-14	88-09	Predefined Type for Slow Protocol Frames
15	03	OAM

The destination address (Bytes 1-6) might be also set to the programmed MAC address, but type and subtype field have to be set to the values specified in **Table 28**. As soon as the filter matches, these frames will be extracted out of data stream and not forwarded. As in the general extraction/insertion part, a bit mask is available for the OAM filter as well.

The SOCRATES™-4e generates OAM frames with the header as specified in **Table 28**. The own MAC address will be used as source address.

Processing of the OAM frames will be done in the IDC. The functional set is tbd.

4.2.4 CPE Packet Application, PHY Mode

This chapter shows and describes the functional blocks of the xTC involved for realizing the CPE Packet Application, PHY Mode. The following figure highlights these blocks. Note that this section only contains the description of new blocks which are not covered in **Chapter 4.2.2**.

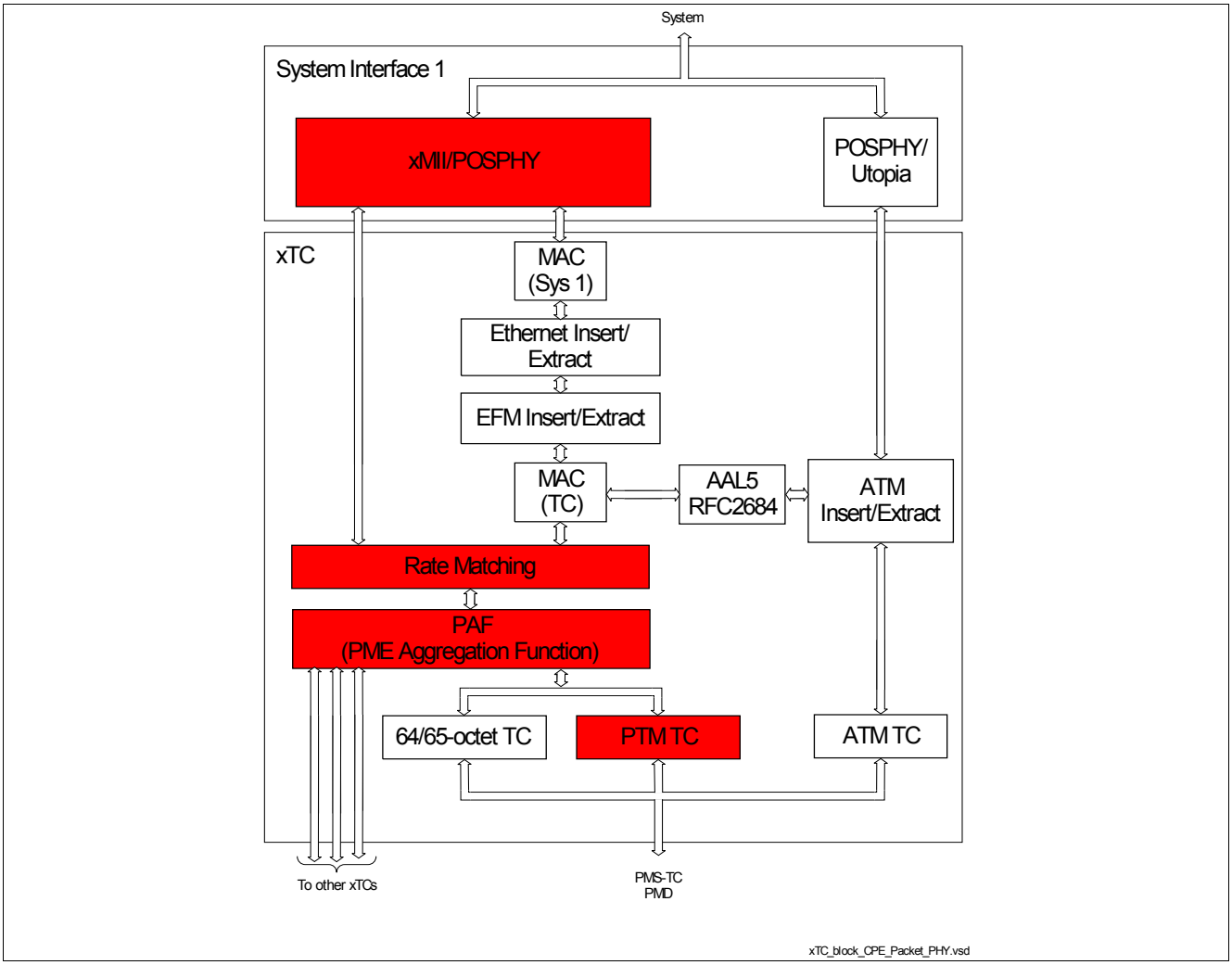


Figure 36 xTC Block Diagram, CPE Packet PHY Mode

The only difference to the CPE EFM PHY mode application (see **Chapter 4.2.2**) is the use of the PTM TC instead of the 64/65 octet TC.

4.2.4.1 PTM TC

The HDLC protocol unit provides protocol handling for up to 4 protocol entities. The protocol unit implements 2 basic modes, which can be programmed independently for each protocol entity: HDLC bit- or octet- synchronous. In both modi, it can be selected whether Address Control Field (ACF) compression is enabled or not. The mode-specific handling is depicted in **Figure 37** and **Figure 38**.

Data Format from System Side

It is expected that entire Ethernet Frames or fragments arrive at the system side of the PTM TC.

Data Format at Line Side

The data format at the line side depends the selected compressions as shown in **Figure 37** and **Figure 38**.

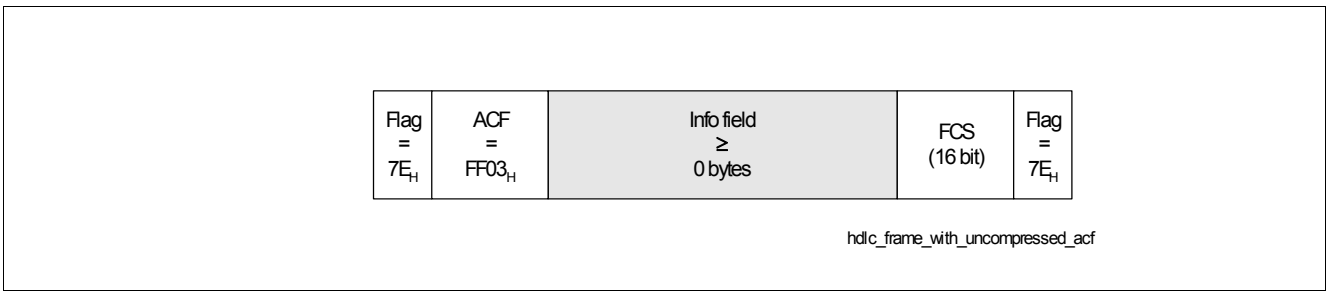


Figure 37 HDLC Frame without Stuffing, uncompressed ACF

Figure 38 HDLC Frame without Stuffing, Compressed ACF Field

Order of Bit Transmission

All octets are transmitted least significant bit first (in contrast to ATM-TC, where MSB-first is applied). For example, the first bit of the control field, equal 03_H, that is transmitted has the value ’1’. The FCS (CRC) will be sent MSB first.

FCS Calculation and Check

The Frame Check Sequence field FCS is 16 bits (two octets).

The transmitter calculates the FCS field over all bits of ACF and the to-be-HDLC-encapsulated packet, not including any bits (bit-synchronous mode) or octets (octet-synchronous mode) inserted for transparency. This also does not include the flag sequences nor the FCS field itself. Accordingly, the FCS field evaluation is done in the receiver after destuffing.

- FCS formula: $1+x^5+x^{12}+x^{16}$

In case of CRC error the receiver will forward the HDLC-decapsulated packet to the packet based system side marked with error.

FCS Calculation and check can be turned off (applies always to both TX and RX direction)

Bit Stuffing for Bit-Synchronous Mode

After FCS computation, the transmitter examines the entire frame between the two Flag Sequences. A ’0’ bit is inserted after all sequences of five contiguous ’1’ bits to ensure that a Flag Sequence is not simulated. On reception, prior to FCS computation, any ’0’ bit that directly follows five contiguous ’1’ bits is discarded by the receiver.

Octet Stuffing for Octet-Synchronous Mode

On transmission, the characters $7D_H$ and $7E_H$ in the data between opening and closing flag are replaced by a two-octet sequence consisting of the Control Escape octet $7D_H$ followed by the original octet exclusive-or'd with 20_H.

The receiver examines the frame between two flags and replaces a two-octet sequence with first byte equal $7D_H$ by the 2nd octet of the sequence exclusive-or'd with 20_H.

The above mentioned transparency procedure will be applied to the octets $7D_H$ and $7E_H$ only (extended transparency as mentioned in ISO 3309 Chapter 4.5.3 is not supported).

Note that the SHDSL standard ITU-T G.991.2 (2004) only supports Octet Stuffing and no Bit Stuffing.

Flag Insertion and Detection

All HDLC frames start and end with the flag sequence 7EH. The receiver continuously hunts for the flag for frame synchronization. A single flag can be used as both the closing flag for one frame and the opening flag for the next frame.

The usage of shared opening and closing flag can be enabled in the transmitter by the user per protocol entity and is automatically detected in the receiver. Two consecutive flag sequences constitute an empty frame, which is silently discarded, and not counted as an FCS error in the receiver.

Abort Sequence Insertion

The transmitter generates an Abort Sequence if the to-be-encapsulated packet is marked with error.

The value of the inserted abort sequence depends on the selected interframe time fill pattern and the used mode bit synchronous - 7 ’1’, octet synchronous octet sequence $7D_H$ $7E_H$.

The receiver detects an aborted frame in the bit-synchronous mode if 7 or more ‘1’s are received. The receiver detects an aborted frame in the octet-synchronous mode if a control escape octet $7D_H$ is followed by the closing flag $7E_H$. Such invalid frames shall be silently discarded and not counted as an FCS error according to IETF RFC 1162. Instead of discarding - this would require a large amount of buffer capacity - the error marking is applied when the decapsulated packet will be forwarded to the system side.

Interframe TIme Fill

If no data from system side is available for the transmitter an interframe time filling pattern of $7E_H$ (so called flag) will be inserted.

ACF Insertion and removal

The ACF $FF03_H$ will be prepended for packets coming from the system side and will be removed in the other direction. Additionally the contents of the ACF to be transmitted field might be modified. In RX direction, the contents of the ACF field of the last received frame can be read out. As soon as there is a change in the received ACF field, a respective notification (if enabled) informs the external controller about it.

The ACF can also be compressed. In this case no ACF field will be inserted in TX direction and nothing will be removed in RX direction.

Invalid Frames

In RX direction the following frames are considered to be invalid:

- Frames shorter than 64 bytes (incl. ACF and FCS)
- Escape Sequence followed by a flag (octet synchr.)
- Frames which contain not valid escape sequences (octet synchr.)
- Frames where the length is not an integral number of octets (after ’0’ extraction -bit synchr.)

Frame Length Supervision

Frames which are too short are marked as invalid frames. Frames which are too long will be marked and counted as well

Errored Frames

An errored frame is a received frame with CRC error.

MIB Support

The following counters are maintained in TX direction:

- Number of transmitted packets (total)
- Number of received packets from system interface
- Number of aborted packets due to error marking (already received with an error from system interface

The following counter are maintained in RX Direction:

- Number of received packets (total)
- Number of received packets with CRC Error
- Number of discarded packets due to RX buffer Overflow
- Number of received aborted packets
- Number of received oversized packets
- Number of received invalid frames
- Number of forwarded packets to next instance (total)
- Number of forwarded packets with Error signal asserted

4.2.5 CPE Packet Application, MAC Mode

This chapter shows and describes the functional blocks of the xTC involved for realizing the CPE Packet Application, MAC Mode. The following figure highlights these blocks.

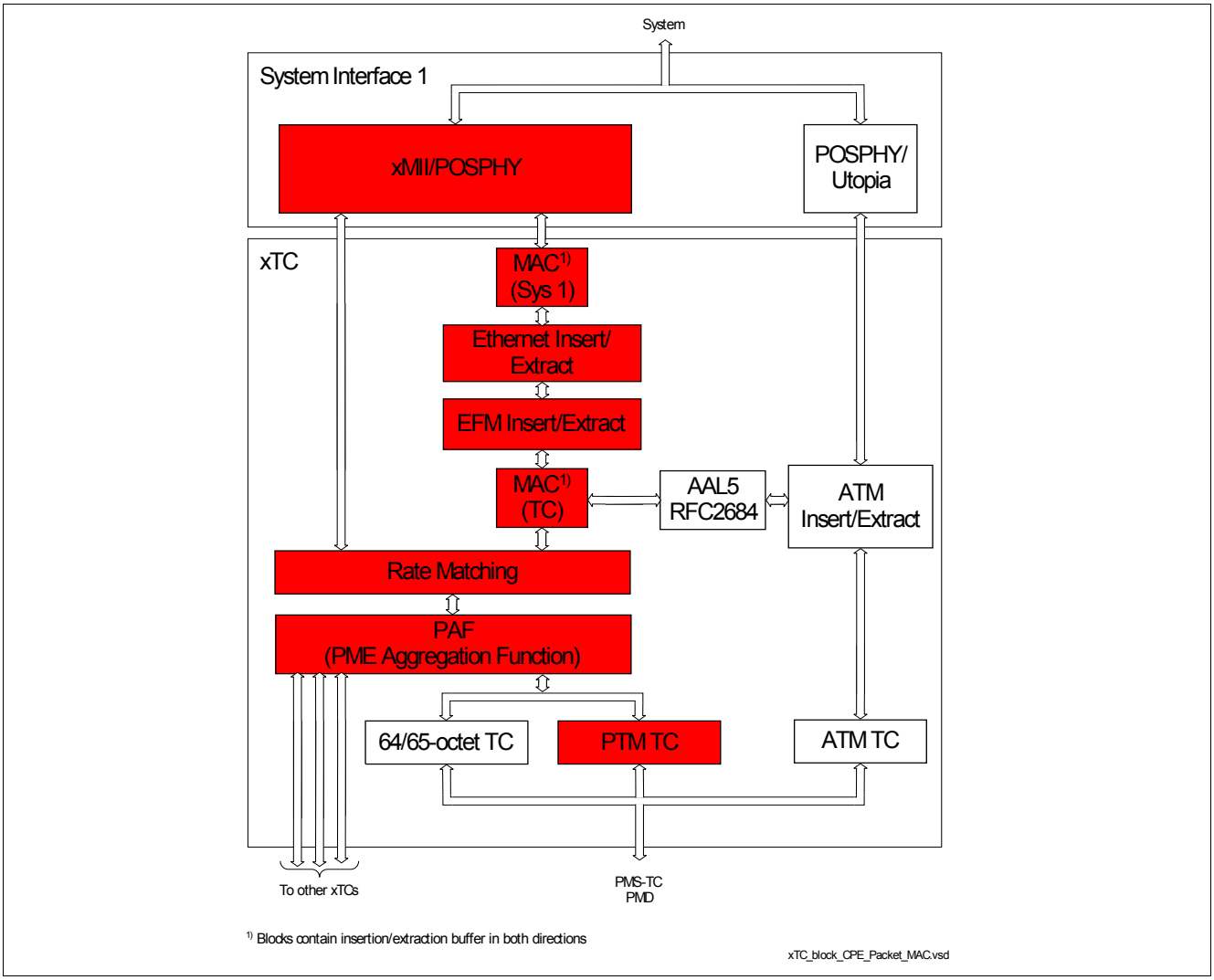


Figure 39 xTC Block Diagram, CPE Packet MAC Mode

Refer to the previous 3 chapters (**Chapter 4.2.2**, **Chapter 4.2.3** and **Chapter 4.2.4**) for the description of the functionality of the different blocks.

4.2.6 ATM Application (CO and CPE)

This chapter shows and describes the functional blocks of the xTC involved for realizing the CPE and CO ATM Application. The following figure highlights these blocks.

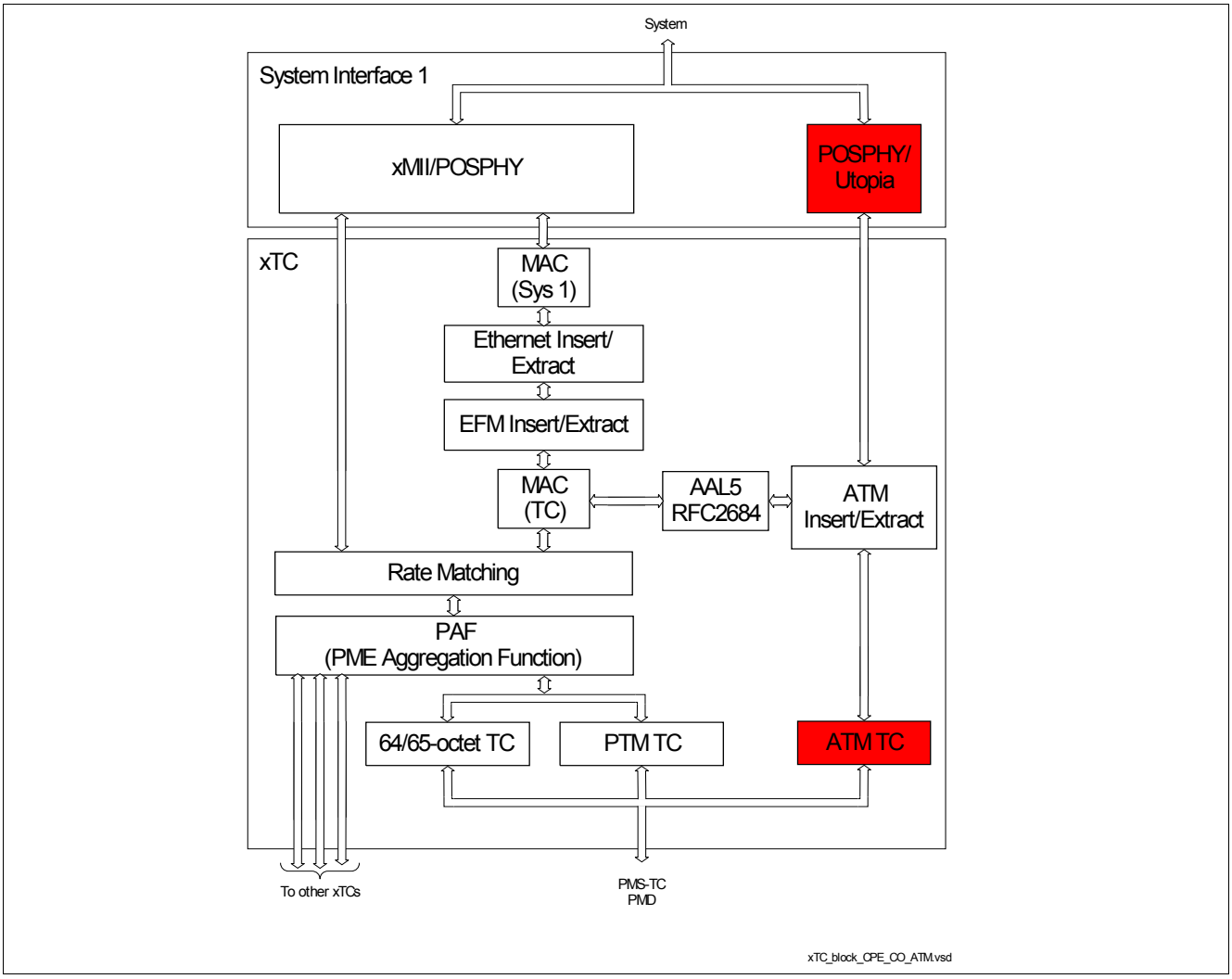


Figure 40 xTC Block Diagram, ATM Mode

There is 1 additional instance within the SOCRATES™-4e which is not related to any data flow and not shown in the above picture. This is an ATM Insert/Extract instance, it is connected to the System Interface 1 (UTOPIA/POS-PHY mode). This instance can extract ATM cells based on filter settings coming from the UTOPIA/POS-PHY interface and is also able to insert cells towards the System Interface 1. This block can be used for remote device configuration.

4.2.6.1 ATM TC

ATM cell mapping and demapping according to ITU-T G.991.2 and ETSI TS 101 524 V 1.2.1 can be used at different SHDSL data rates. Single bearer (entire SHDSL payload dedicated to ATM) and Dual Bearer mode (entire SHDSL payload split up between 2 bearer) applications are supported. Up to 4 protocol entities in total are available (1 per SHDSL link). Within an SHDSL frame a fraction can have a bandwidth of N x 64kbit/s (N = 1..89) + i x 8kbit/s (i = 0..7, with i limited to 0 at N = 89) The maximum bandwidth of a physical channel is 5696 kbit/s (N = 89 and i = 0), 8 kbit granularity for the ATM-TC sublayer is supported. Applications like E1 over ATM is therefore supported.

For Four-wire/M-pair applications the bandwidth can be increased according to the number of ports.

ATM Cell Mapping Function

- Write ATM cells from System Interface 1 Port to transmit buffer of the assigned ATM-TC Protocol Entity

- Hold up to 2 ATM cells in transmit buffer of ATM-TC Protocol Entity and generate back pressure to the assigned System Interface 1 Port when this buffer has reached its full level
- Reading octets from Transmit Buffer of ATM-TC Protocol Entity
- Cell Rate Decoupling: Idle/Unassigned Cell insertion
- Cell Payload Scrambling
- HEC Generation
- Statistics counter event generation (see appendix of the User's Manual Programmer's Reference)

Cell Rate Decoupling: Idle/Unassigned Cell Insertion

When the transmit buffer of the ATM-TC Protocol Entity is empty, idle or unassigned cells are transmitted to provide cell rate decoupling.

Idle cells are transmitted as defined in the ITU-T I.361. Unassigned cells can be inserted, as defined in the B-ISDN UNI and NNI physical layer generic criteria Bellcore TR-NWT-001112.

The 4 MSBs of header octet 1 and the 4 LSBs of header octet 4 are programmable for unassigned cells. All other header bits will be 0.

If idle cell insertion according to ITU-T I.361 or ITU-T I.432.1 is selected, the 4 LSBs of header octet 4 are set to 0001_B and the 4 MSBs of header octet 1 are set to 0000_B.

Table 29 Idle/Unassigned Cell Header

octet 1	GFC[3:0]/VPI[11:8] = programmable	VPI[7:4] = 0000B	
octet 2	VPI[3:0] = 0000B	VCI[15:12] = 0000B	
octet 3	VCI[11:4] = 0000_0000B		
octet 4	VCI[3:0] = 0000B	PTI[2:0] programmable	CLP programmable
octet 5	HEC		
octet 6	0110_1010B		
.	.		
octet 53	0110_1010B		

If unassigned cell insertion at the NNI or uncontrolled UNI according to ITU-T I.361 is desired, the 4 LSBs of header octet 4 should be set to XXX0 and the 4 MSBs of header octet 1 should be set to 0000. For X any value is allowed.

The S(H)DSL standards ITU-T G.991.2 and ETSI TS 101 524 V 1.2.1 only support idle cell insertion for cell rate decoupling.

The payload of idle or unassigned cells consists of the same octet which is repeated 48 times. For ITU-T I.432.1 compliant idle cells, the payload octet is set to 0110_1010B. The pre-assigned values of the information field of all unassigned cells are for further study (ITU-T I.361) and the payload octet can be set by the user.

Cell Payload Scrambling

ITU-T I.432.1 recommends the self-synchronizing scrambler polynom $x^{43}+1$ for payload scrambling. The scrambler function is implemented in the device. It can be disabled which applies to entire ATM TC (all links of SOCRATES™-4e.)

HEC Generation

The HEC generation is implemented according to ITU-T I.432.1 using the generator polynomial $x^8 + x^2 + x + 1$. To significantly improve the cell delineation performance in the case of bit-slips it is recommended that

- The check bits are added (modulo 2) to an 8-bit pattern (coset) before being inserted in the last octet of the header
- The recommended pattern is “0101 0101" and automatically used by the transmitter
- The receiver must subtract (equal to add modulo 2) the same pattern from the 8 HEC bits before calculating the syndrome of the header.

As an example, if the first 4 octets of the header were all zeros the generated header before scrambling would be “00000000_00000000_00000000_00000000_01010101”. The starting value for the polynomial check is 0s (binary).

Cell Demapping Functions

- Cell Delineation
- HEC Check: Header error detection and correction
- Cell Payload Descrambling
- Idle or Unassigned Cell Deletion
- Write up to 4 cells, except of 5th header octet, to Receive Buffer of ATM-TC Protocol Entity
- Read cells from Receive Buffer of ATM-TC Protocol Entity, insert 5th header byte with value 00H and forward to the assigned System Interface 1 Port
- Statistics counter event generation (see appendix of the User’s Manual Programmer’s Reference)

Cell Delineation

The cell delineation algorithm is implemented according to the ITU-T Recommendation I.432.1.

Detection of “Out of Cell Delineation” (OCD) anomalies and “Loss of Cell Delineation” (LCD) defect is supported whenever the SYNC state is left or entered. The generation of interrupts due to LCD defects can be enabled. It is also possible to see the current state of the cell delineation FSM (Finite State Machine) on demand.

As octet boundaries are available within the receive physical layer prior to cell delineation, the cell delineation process is performed octet by octet in the HUNT state. As long as the cell delineation is not in the SYNC state, received octets are discarded.

The ALPHA and DELTA parameters influence the robustness of the algorithm against false misalignment due to bit errors (ALPHA) and false delineation in the re- synchronization process (DELTA). The used values are common for all protocol entities according to the ITU-T I.432.1 recommendation:

For a SDH-based Physical Layer, ALPHA = 7 and DELTA = 6.

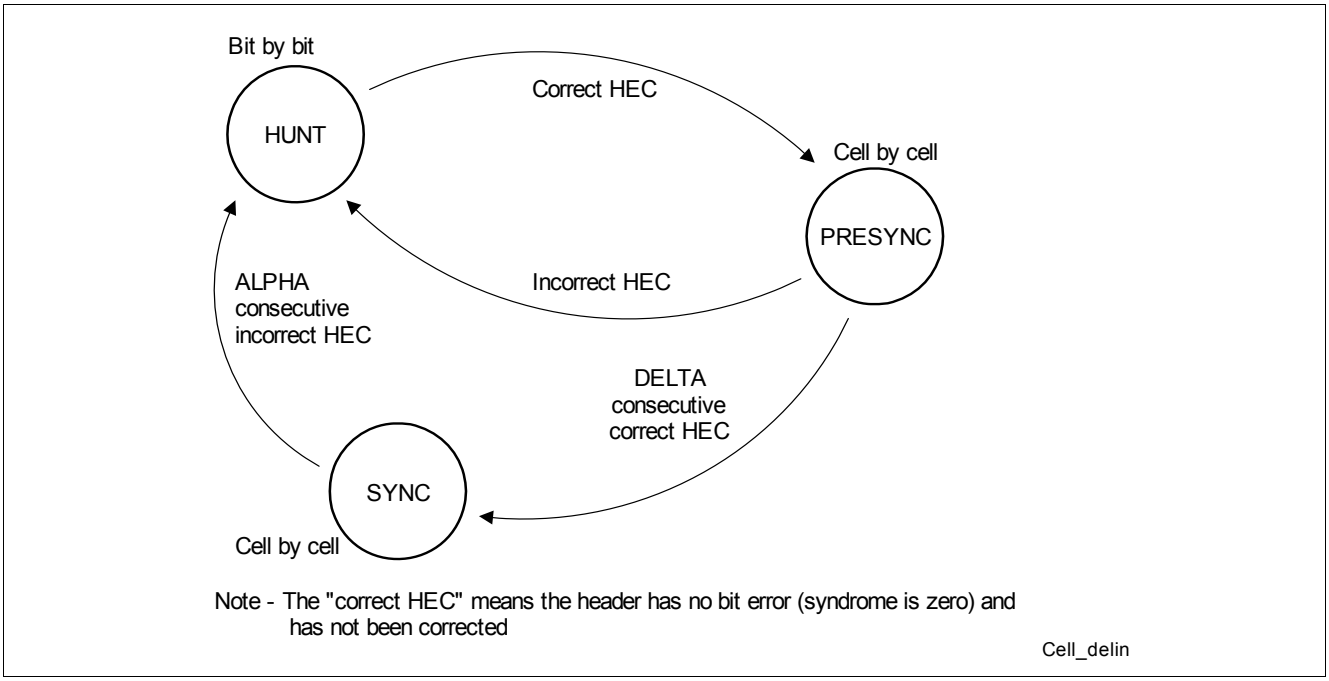


Figure 41 Cell Delineation State Diagram (accdg. to ITU I.432)

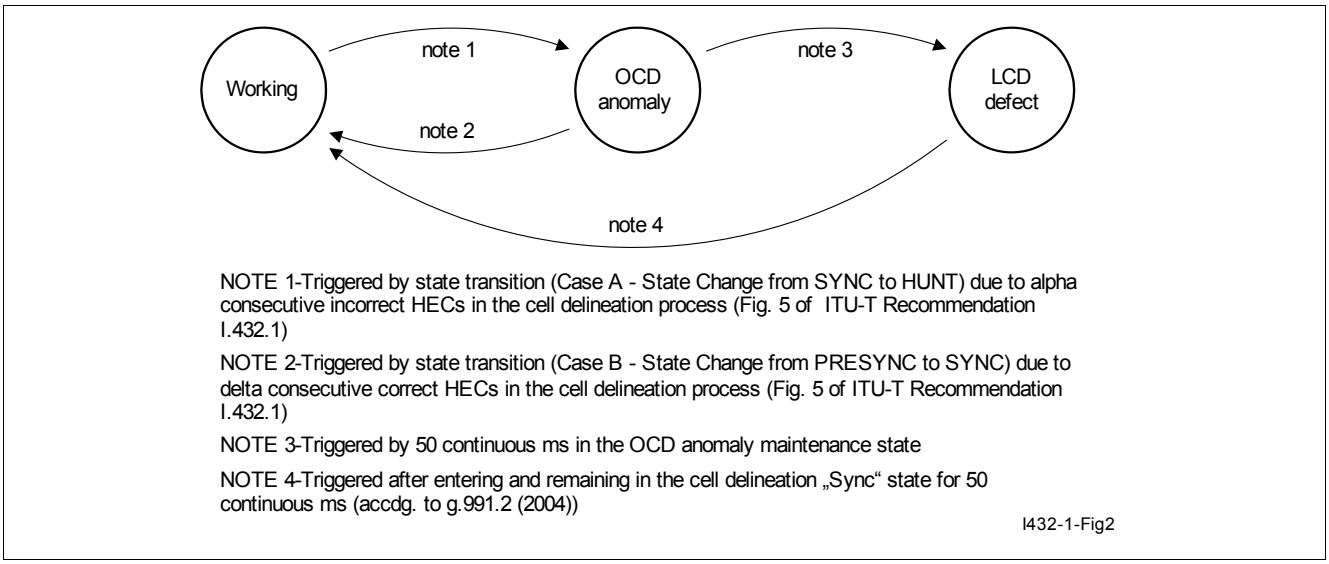


Figure 42 Maintenance State Transition Diagram

Figure 42 depicts the Maintenance State Transition Diagram for Cell Delineation Events according to Figure 3 / I.432.3 (02/99). The Loss of Cell Delineation (LCD) state is entered whenever the Out of Cell Delineation (OCD) state is continuously active for more than 50 ms (as recommended by ITU-T I.432.1).

For each protocol entity, a separate timer is available. The global preload value is 50 ms. After expiration of each timer, an “lcd_start” event is generated.

The timer is started at the transition from SYNC to HUNT-state. After expiration LCD state is entered. Whenever the SYNC state is entered before the timer expires, the timer is reset.

The transition from LCD to Working state follows the same procedure. If after the LCD state the SYNC state is entered again, the timer is started and after expiration the maintenance state machine is in working state again. In parallel an “lcd_end” event is generated. If synchronization is lost again during the timer period, LCD state is reentered and the timer is reset.

HEC Checking

According to the HEC algorithm, cells are discarded when an error is detected.

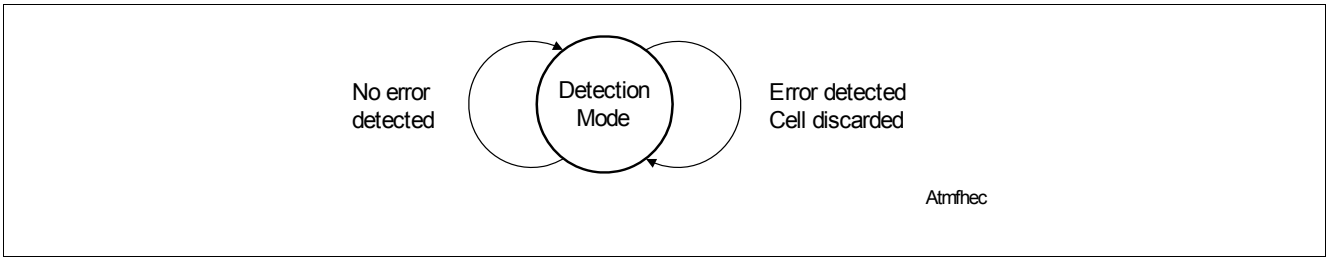


Figure 43 HEC Error Detection according to ATM Forum

The SOCRATES™-4e supports 2 possible mechanisms for forwarding the HEC byte to the next functional block.

- Forward HEC Byte
- Replace HEC Byte by UDF field (user defined field - contents of UDF programmable)

IMA Support

The following picture shows a typical IMA application with SOCRATES™-4e, the number of used SOCRATES™-4e depends on the application.

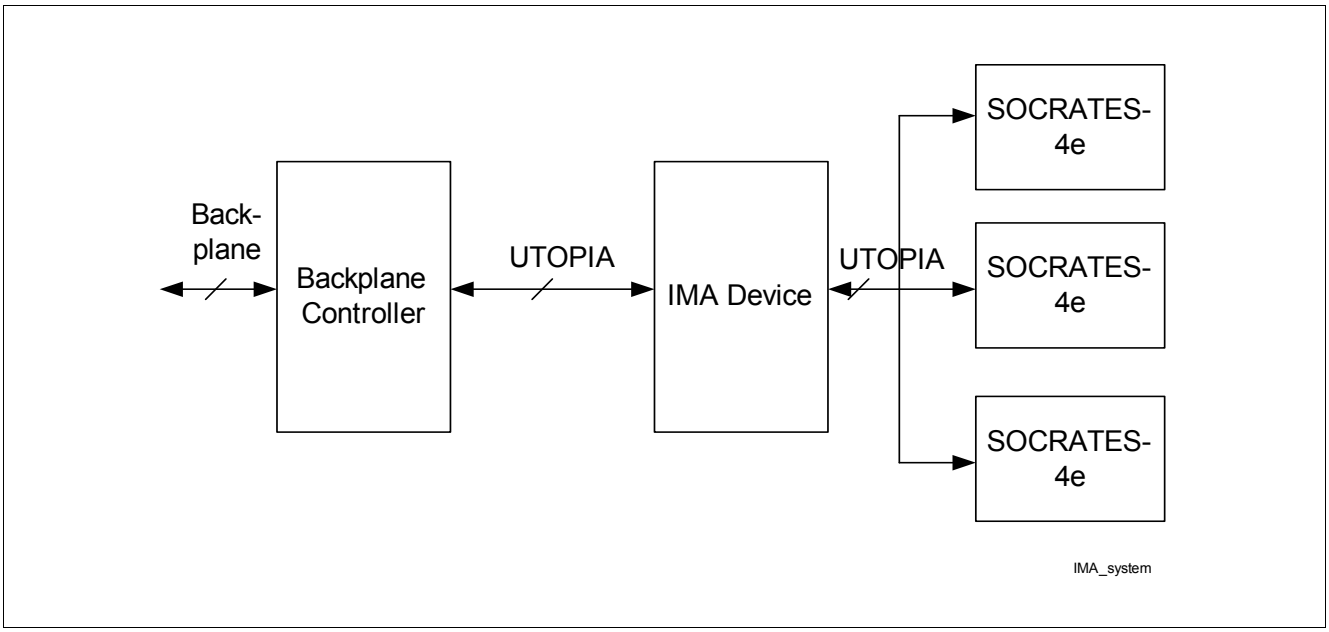


Figure 44 IMA Application with SOCRATES™-4e

In case of IMA applications no ATM cells with HEC errors may be discarded, they are still needed by the IMA layer for synchronization purposes. The SOCRATES™-4e supports these kind of application (IMA functionality realized outside of the SOCRATES™-4e), in ATM cells with HEC errored cells bit 7 of the HEC Byte is set to “1” . All non errored cells have this bit set to 0 in this mode. Sending an UDF in ’IMA Mode’ is not supported.

It is furthermore expected that no idle cells will be used for rate matching, so called filler cells have to be used for this purpose.

Cell Payload Descrambling

ITU-T I.432.3 recommends the self-synchronizing scrambler $x^{43}+1$ for payload scrambling. The self-synchronizing scrambler function is implemented in the device. It can be disabled, which applies, as for the scrambler, to the entire SOCRATES™-4e.

Idle/Unassigned Cell Deletion

Idle or unassigned cells are inserted for cell rate decoupling and need to be removed by the RX part of the ATM TC. The SOCRATES™-4e supports both Idle and Unassigned cells for this purpose. It can be selected (applies to the entire chip) whether Idle or Unassigned cell shall be interpreted by the ATM-TC as cell rate decoupling cells and thus be removed.

MIB Counters

Each ATM TC maintains a set of MIB counters (separate RX and TX counter). For details refer to the User's Manual System Description.

4.2.7 CPE Interworking Scenarios

This chapter shows and describes the functional blocks of the xTC involved for realizing the interworking function on the CPE side. In CPE interworking applications an interworking between ATM cells on the SHDSL line and Ethernet packets on the system side takes place. The following figure highlights the blocks needed for this functionality.

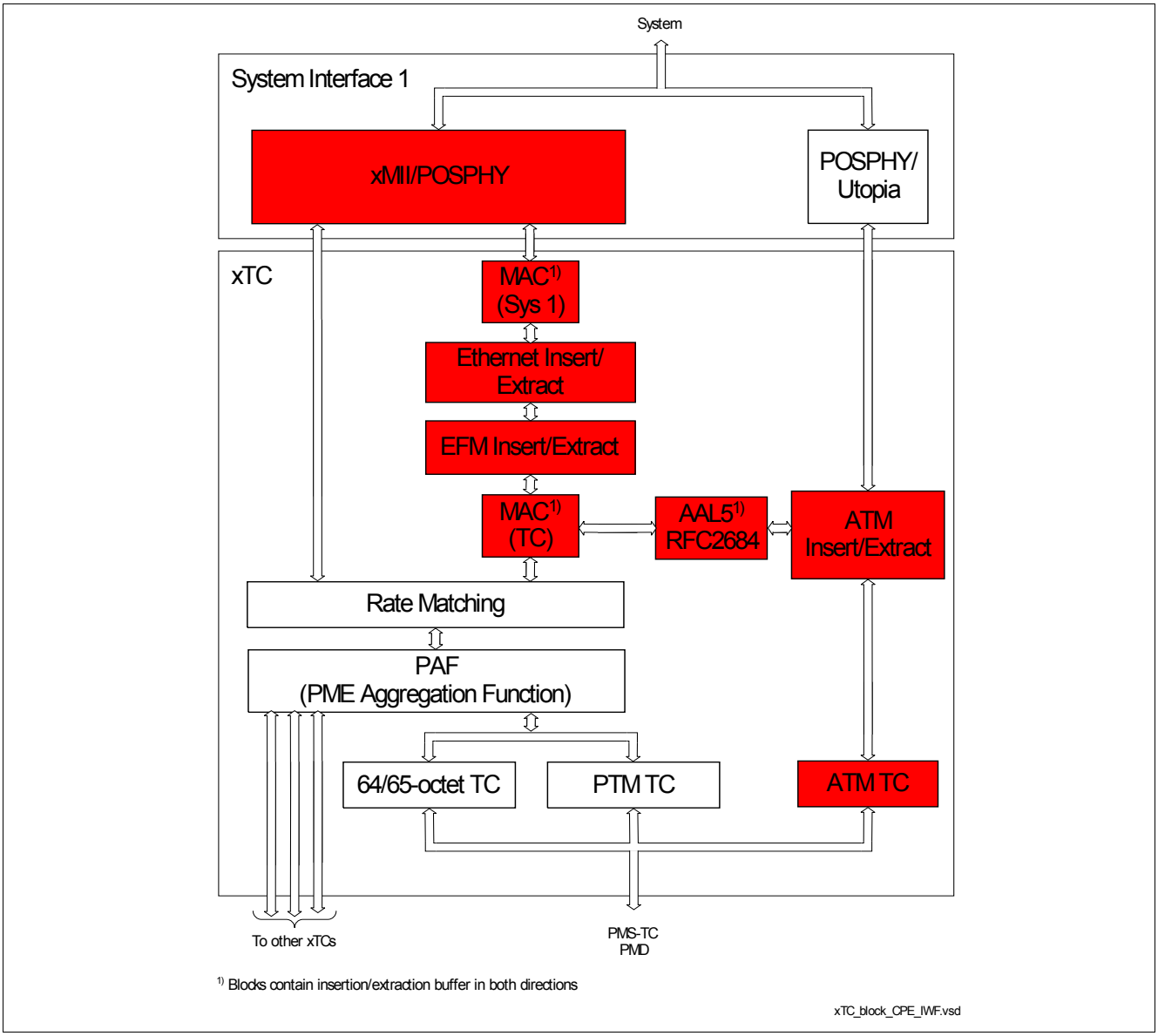


Figure 45 xTC Block Diagram, CPE Interworking Mode

The function of the blocks ATM TC, MAC and Ethernet/EFM Insert/Extract is already described in previous chapter, refer to **Chapter 4.2.6.1**, **Chapter 4.2.3.1** and **Chapter 4.2.3.2** for details.

4.2.7.1 AAL5/RFC2684 Block

This block consists out of 2 major parts. The AAL5 (ATM Adaption Layer 5) does the interworking between packets and cells, the RFC part adds/remove additional information to/from the packet in order to allow classification of the packets.

4.2.7.2 AAL5 Functionality

The AAL5 functionality (CPCS and SAR) is implemented according to the AAL5 standard ITU-T I.363.5.

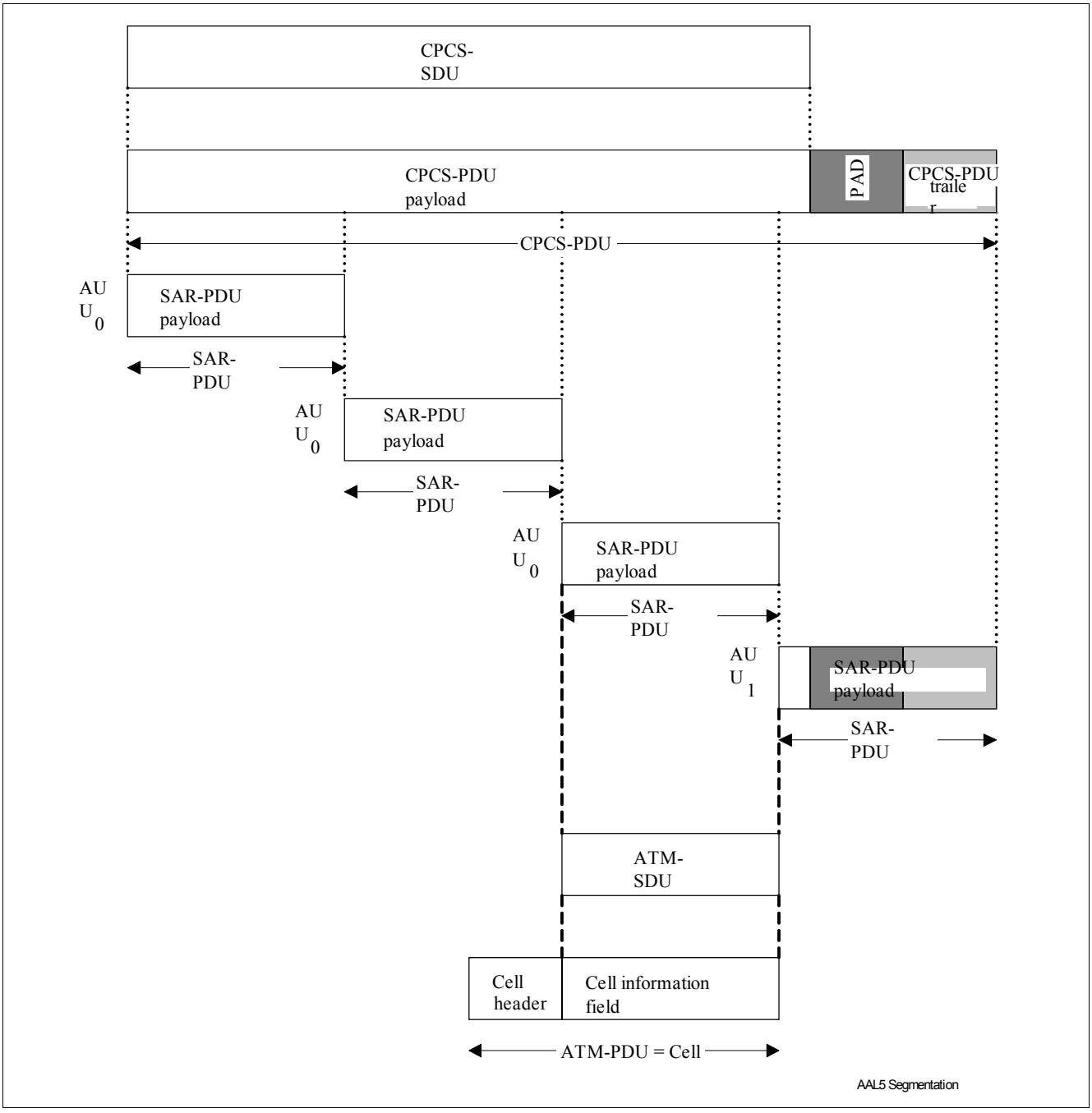


Figure 46 Generic View of AAL Segmentation

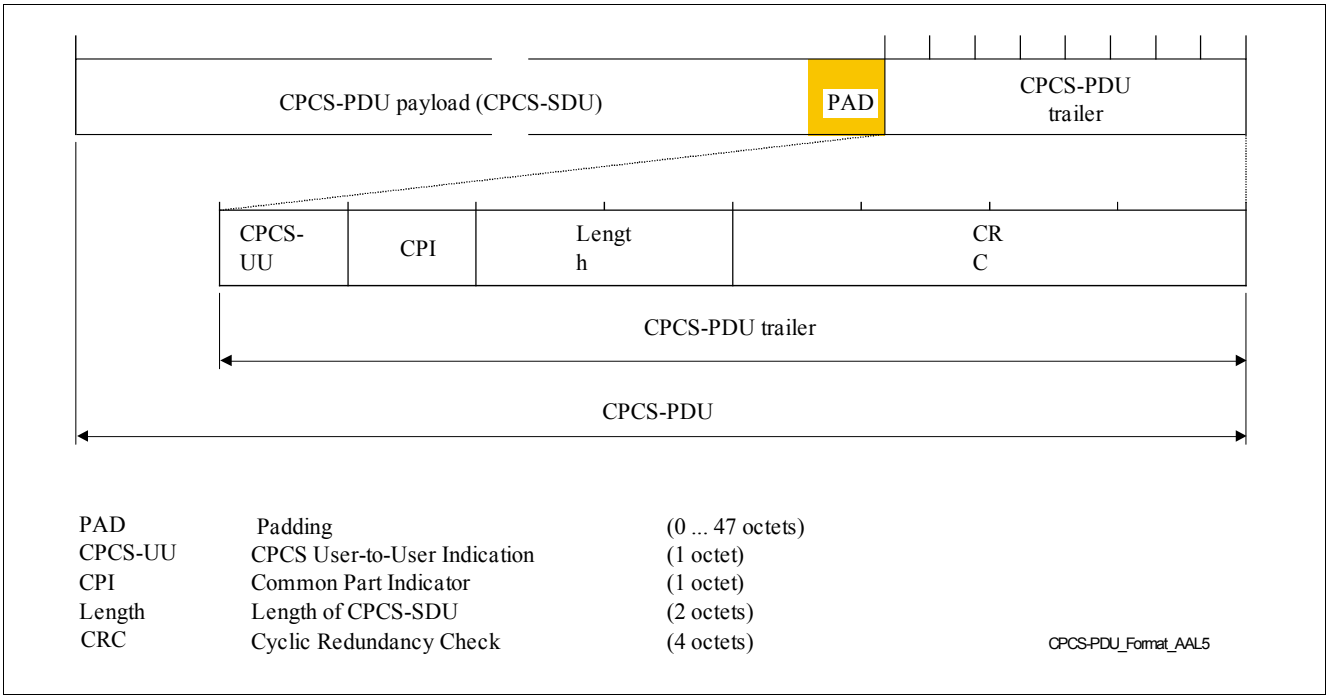


Figure 47 CPCS PDU Format AAL5

The CPCS part (Common Part Convergence Sublayer) adds the CPCS PDU trailer. This trailer is always located at the last 8 byte of the last SAR PDU of the CPCS PDU. Therefore, a padding field provides for a 48-octet alignment of the CPCS-PDU. The CPCS-PDU trailer together with the padding field and the CPCS-PDU payload comprise the CPCS-PDU.

The CPI field is used to align the CPCS-PDU trailer to 64 bits and is coded as 00H by the transmitter. The receiver doesn't evaluate this field and simply removes it.

The sizes and positions of fields for the CPCS-PDU structure are given in **Figure 47**.

CPCS functions per CPCS-PDU are handled in following way

Preservation of CPCS PDU

This function provides for the delineation and transparency of CPCS-SDUs. The CPCS-PDU payload is used to carry the CPCS-SDU. This field is octet aligned and can range from 1 to max frame length octets in length. If the maximum CPCS-SDU is exceeded, i.e. the to-be-segmented packet is larger than max frame length octets, the Abort function is applied.

Preservation of User to User Information

This function provides for the transparent transfer of CPCS user-to-user information by using the CPCS-UU field. This function is not supported since this field is set to the fixed value 00_H in the transmitter and is discarded without evaluation in the receiver.

Error Detection and Handling

This function provides for the detection and handling of CPCS-PDU corruption. The transmitter supports the calculation of CRC field and length field.

CRC-32 is used to detect bit errors in the CPCS-PDU. The CRC field is filled with the value of a CRC calculation which is performed over the entire contents of the CPCS-PDU, including the CPCS-PDU payload, the PAD field, and the first four octets of the CPCS-PDU trailer.

Length field is used to encode the length of the CPCS-PDU payload field. The length is binary encoded as number of octets and generated by an 11-bit octet-counter. A Length field coded as zero is used for the abort function.

Corrupted CPCS-SDUs are not discarded by the receiver and forwarded as packets with error indication. They are treated as corrupt if

- Received CPCS-SDU length and CPCS-PDU length field doesn't match
- Received CPCS-SDU length exceeds the user-configurable value for the maximum permitted length Max_SDU_Deliver_Length (configurable in the range of 1..2047 octets, supervision is always active)
- CPCS-PDU length field is equal zero
- Received CPCS-CRC and calculated CRC doesn't match

If the CPI field value is not valid the CPCS-SDU will not be treated as corrupt (CPCS receiver doesn't evaluate the CPI field, it simply discard the field).

A reassembly time-out functionality is not supported since CPCS and SAR sublayer are working in streaming mode without storing a large amount of data.

Abort Function

This function provides for the means to abort a partially transmitted CPCS-SDU. This function is indicated in the length field by value zero. The transmitter applies the Abort function if

- Received CPCS-SDU exceeds the user-configurable value for the maximum permitted length (configurable in the range of 1..2047 octets, not off-switchable)
- CPCS-SDU has been received with error indication

Padding

A padding function provides for 48-octet alignment of the CPCS-PDU trailer. Between the end of the CPCS-PDU payload and the CPCS-PDU trailer, there will be from 0 to 47 unused octets. These unused octets are called the Padding (PAD) field. They are strictly used as filler octets and do not convey any information. Padding octets are coded with any value by the transmitter and removed by the receiver. This padding field complements the CPCS-PDU (including CPCS-PDU payload, padding field, and CPCS-PDU trailer) to an integral multiple of 48 octets.

Handling of Congestion Information

This function provides for the passing of congestion information between the layers above the CPCS and the one below in both directions. This function is not supported since all generated AAL5 ATM cells of one connection get identical ATM headers (except LSB of the PT field) by the transmitter and ATM headers are simply removed by the receiver.

Handling of Loss Priority Information

This function provides for the passing of cell loss priority information between the layers above the CPCS and the one below in both directions. This function is not supported since all generated AAL5 ATM cells of one connection get identical ATM headers (except LSB of the PT field) by the transmitter and ATM headers are simply removed by the receiver.

SAR Sublayer Functions and Coding

The following figure illustrates the mapping of SAR blocks into ATM cells. The general SAR (Segmentation and Reassembly) function is illustrated in the lower part of **Figure 46**.

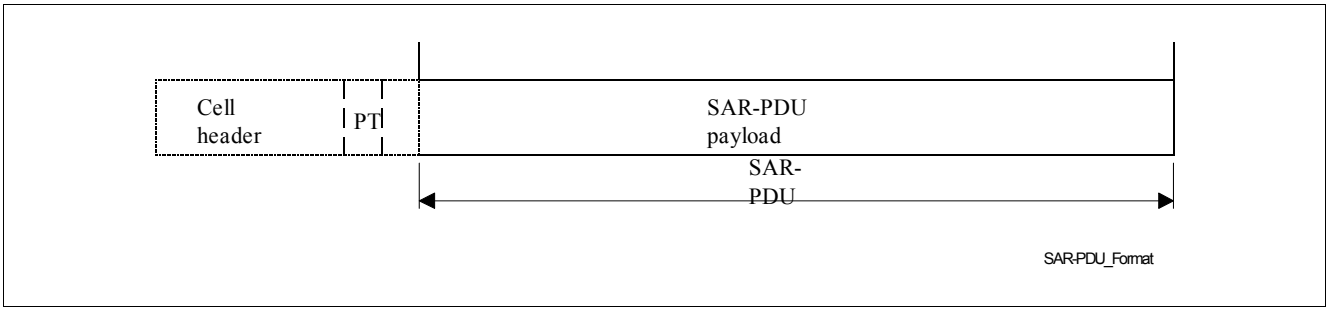


Figure 48 SAR PDU Format

The SAR sublayer functions are performed on an SAR-PDU basis. The SAR sublayer accepts variable length SAR-SDUs which are integral multiples of 48 octets from the CPCS and generates SAR-PDUs containing 48 octets of SAR-SDU data.

The SAR sublayer utilizes the ATM-User-to-ATM-User indication (AUU) parameter of the ATM layer primitives to indicate that an SAR-PDU contains the end of a SAR-SDU. The AUU parameter is coded in the least significant bit of the ATM header field PT. A SAR-PDU where the value of the AUU parameter is "1" indicates the end of an SAR-SDU. The value of "0" indicates the beginning or continuation of a SAR-SDU.

The ATM header for each of the AAL5 ATM connections, which will be prepended by the transmitter, is user-configurable (see CROSS REF RFC Section) and common for all cells of this connection. The first 4 octets of the ATM header can be programmed, other header octets have the fixed value zero. After prepending, the transmitter will overwrite the LSB of the PT field with the AUU parameter. The receiver simply removes the ATM header of received cells. Optionally, the received ATM header can be checked against the programmed transmitted header with applied cell discarding in case of mismatch.

Handling of congestion information and loss priority information is not supported due to the ATM header handling.

4.2.7.3 RFC2684 Functionality

RFC2684 specifies multiprotocol encapsulation over AAL5. Two methods are specified, the first method allows multiplexing of multiple protocols over a single ATM virtual connection. This method is referred to as LLC (Logical Link Control) encapsulation.The second method assumes that each protocol is carried over a separate ATM virtual connection. This method is referred to as VC multiplexing.

The SOCRATES™-4e supports both methods. In both cases it is assumed that the traffic to be carried over is Ethernet traffic.

The entire RFC functionality can be bypassed, in this case no specific header will be carried over the link.

LLC Encapsulation

The protocol type is identified by a prefixed IEEE 802.2 LLC header. It is expected that only Ethernet traffic will be carried over, whereas Ethernet is a bridged protocol. This protocol information must be encoded in an LLC header and placed in front of the PDU.

For bridged protocols, the protocol type will be identified by the SNAP header. The presence of a SNAP header must be identified by the LLC header of value of 0xAA-AA-03. The OUI value in the SNAP header must be the IEEE 802.1 organization code 0x00-80-C2. The type of the bridged media itself will be identified by the PID field. Additionally this PID must contain an information whether the original Frame Check Sequence (FCS) is transmitted or not. The SOCRATES™-4e also supports these 2 modes.

Additional padding octets are inserted in order to align the Ethernet Data field to 4 octet boundary.

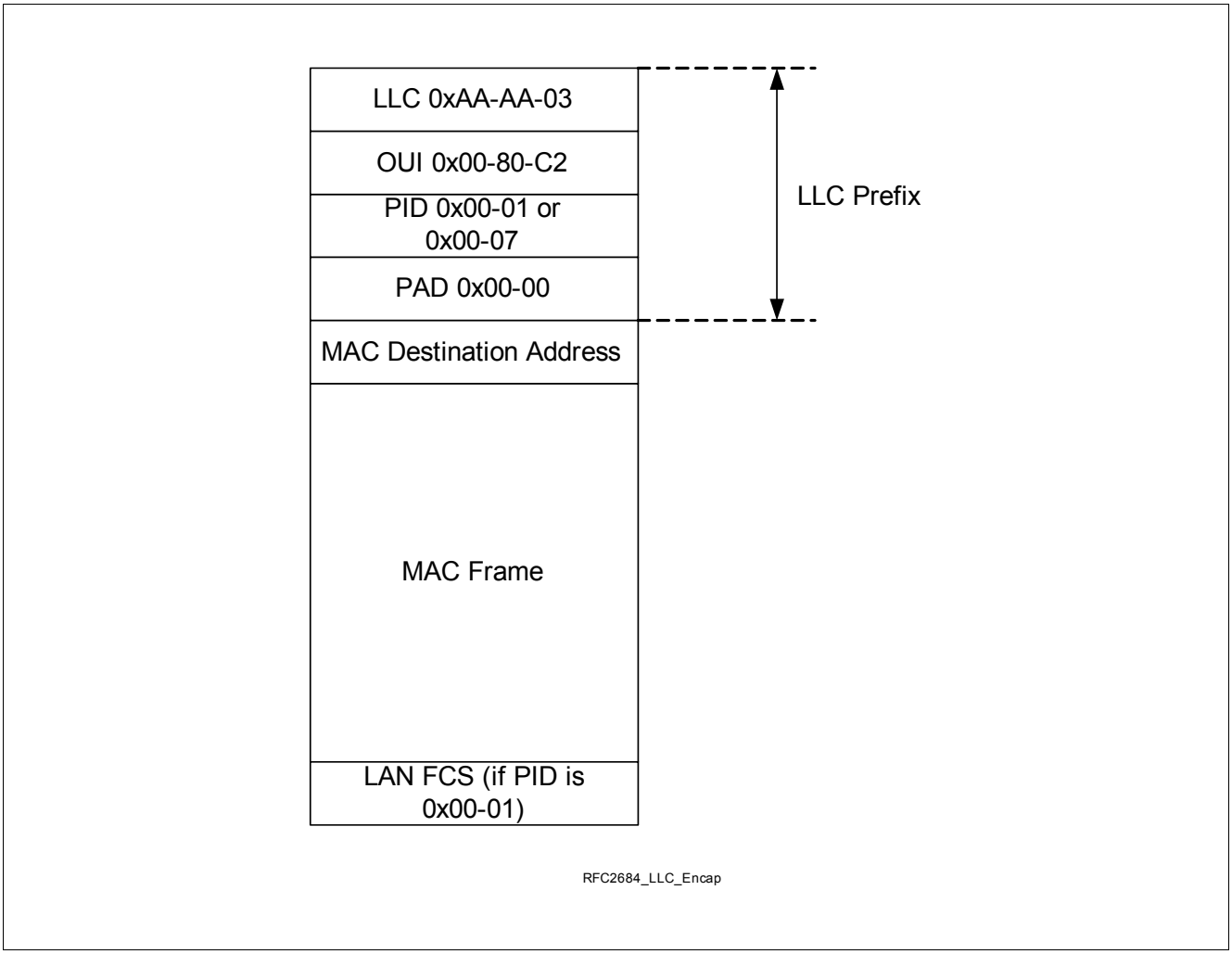


Figure 49 RFC2684, LLC Encapsulation

It can be configured whether the LAN FCS shall be transmitted or not. In receive direction this decision will be done based on the PID (PID = 0x00-07, LAN FCS not present). If the preserved LAN FCS will not be transmitted, any padding bytes will transmitted. If the PID of the received PDU is equal to 0x00-07, it is expected that the minimum MAC frame size is 64 byte in this case.

If LLC encapsulation is selected for a particular link all received Ethernet frames from the System Interface will be encapsulated in the same way and sent over the same VPI/VCI combination after segmentation. This fixed VPI/VCI combination is programmable.

In device receive direction it is expected that data belonging to a PDU will be received in a row. The LLC prefix will be checked. Received reassembled packets with a wrong LLC prefix will be discarded. Afterwards the PID field will be evaluated. Based on the content either the Ethernet Frame CRC will be appended or not. Afterwards the Ethernet frame will be sent to the MAC block.

VC Multiplexing

VC Multiplexing creates a binding between an ATM VC and the type of network protocol to be carried over this VC. As already stated it is expected that the SOCRATES™-4e always carries Ethernet frames in the interworking application. Thus the network protocol is always Ethernet, different type of Ethernet frames or frames with dedicated Ethernet addresses can be carried on different VC.

The SOCRATES™-4e supports up to 8 VPI/VCI combination. The classification of the Ethernet and the mapping to a special VPI/VCI combination in device transmit direction will be done based on a filter. This filter consists out

of 7 bytes, each byte is maskable. An offset parameter determines at which byte position the filter will be applied. This filters works together with a matching table. This matching table has 16 entries and contains the resulting VPI/VCI combination, is freely programmable and needs to be configured before sending any traffic. The following figure illustrates the classification process.

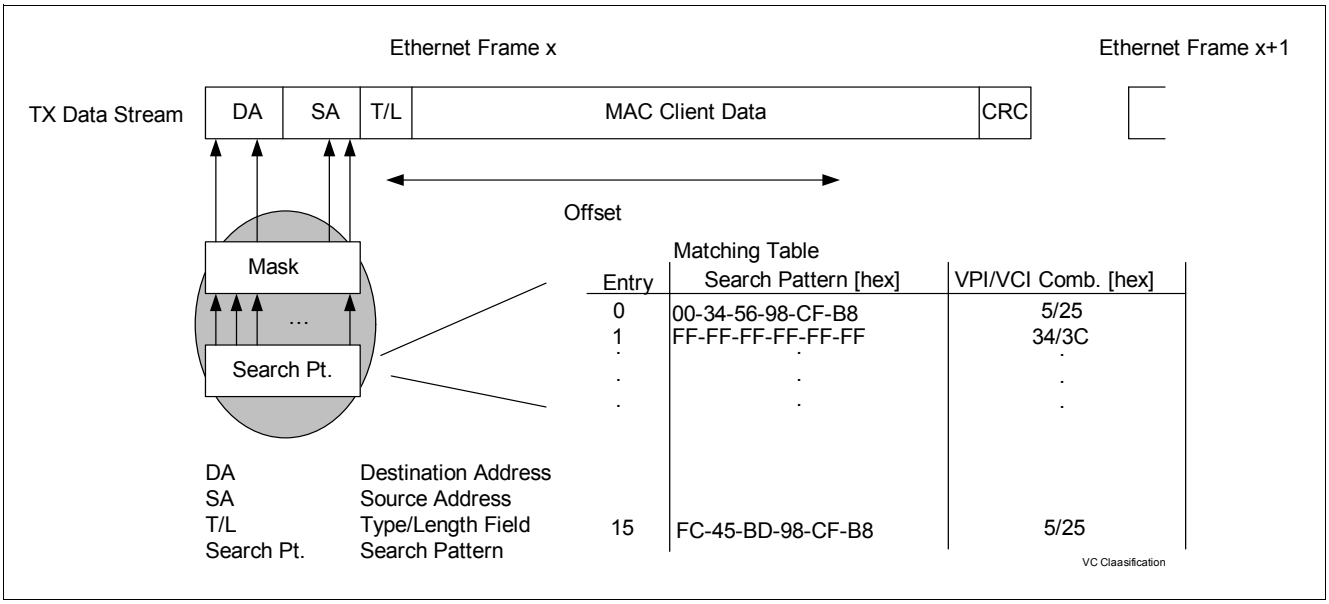


Figure 50 VC Classification

There are 2 different methods in case none of the Search Pattern out of the matching table matches the pattern of the current Ethernet frame. In case of a mismatch the frame can either be discarded or will be sent over a default VPI/VCI. This default VPI/VCI can be either an entry out of the matching table or another fixed defined VPI/VCI combination apart form the Matching table. Additionally the result of the classification returns whether a prefix will be prepended (default value of 0x00-00 for Ethernet protocol) or not and whether the LAN FCS will be carried or not. These information need to be programmed consistently on both sides.

In receive direction supported VPI/VCI combination can be programmed independently, whereas the PTI and the CLP filed can be masked. Each VPI/VCI combination contains additional information about prepended padding and sending of LAN FCS.

The payload of received ATM cells with non-supported VPI/VCI combination will be discarded. The payload of ATM cells belonging to different VPI/VCI combinations will be stored in different queues. Therefore it is not necessary to send data belonging to 1 PDU in an order, different PDU's (related to different VPI/VCI) can be sent in an interleaved session. There is no prioritization between the queues, as soon as 1 Ethernet frame is complete, it will be forwarded to the System Interface 1.

The SOCRATES™-4e contains a set of MIB counters for both encapsulation methods, separated by direction. Refer to the User's Manual Programmer's Reference for further detail.

4.2.7.4 ATM Insert/Extract Block

This block allows to insert/extract ATM cells into the data flows. in total there are 2 independent extraction filter,s they filters are freely programmable.

This module within the dataflow as shown in **Figure 45** can be used for the termination handling of the ATM connection. Since this is the endpoint of the ATM connection, the ATM Insert/Extract block can be used to insert/extract ATM OAM cells (extraction filters to be set accordingly), respective cells can be inserted as well.

Additionally there is another filter available which is freely programmable as well.

ATM OAM Support

This insertion/extraction functionality can be used by an i.e. external ATM OAM processing software stack. Extracted cells are available via a message, ATM cells will be inserted via a message as well. In Insert direction, the ATM cells will be sent into a buffer. The insert block sends these cells out of the buffer with highest priority to the line.

4.2.8 CO Interworking Scenarios

This chapter shows and describes the functional blocks of the xTC involved for realizing the interworking function typically found at the CO side. In CO interworking applications an interworking between ATM cells on the system side and Ethernet packets on the SHDSL side takes place. The following figure highlights these blocks needed for this functionality.

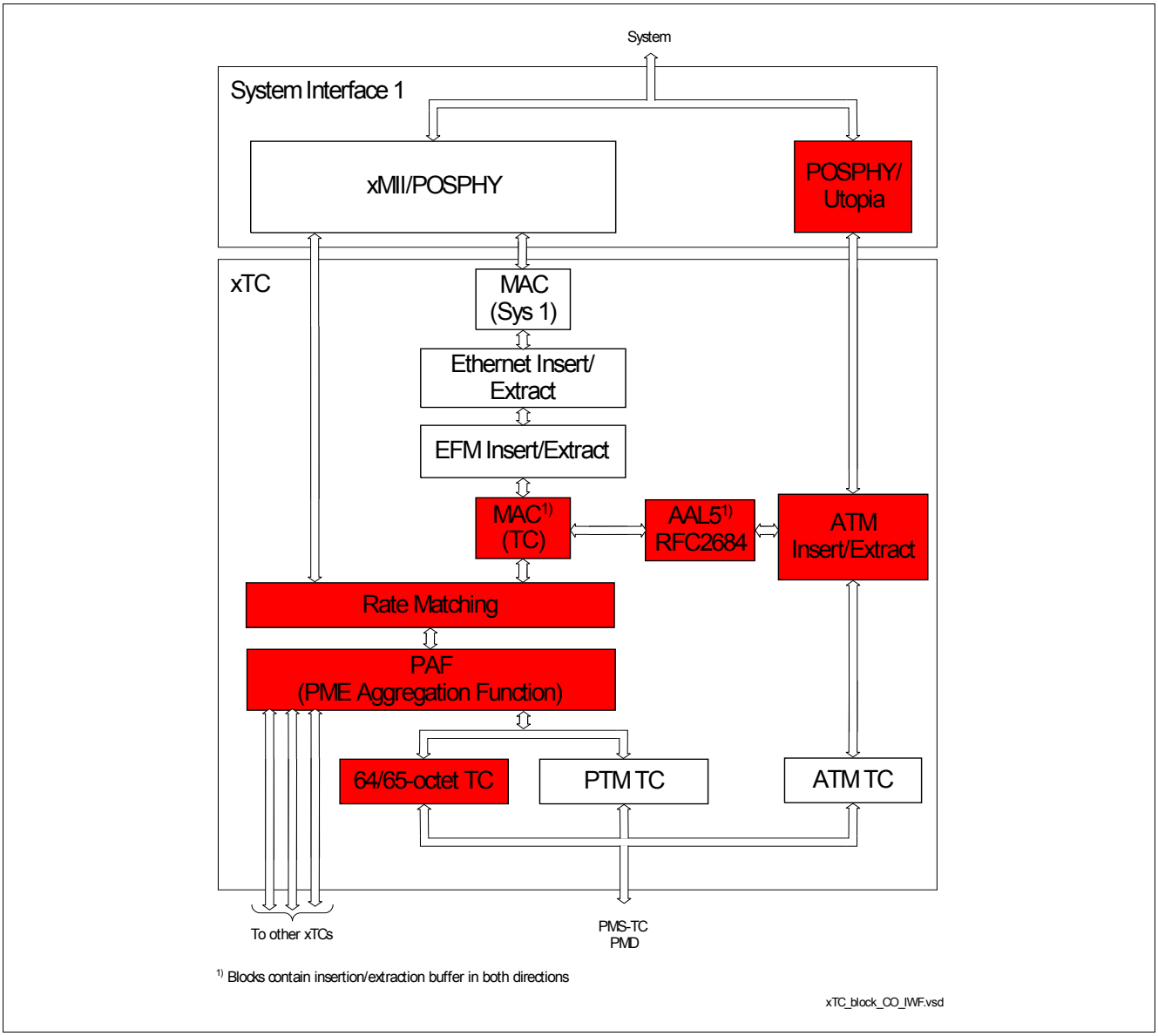


Figure 51 xTC Block Diagram, CO Interworking Mode

The function of the all the blocks involved is already described in previous chapters. Note that PAF functionality can be disabled and the PTM TC might be used as and alternative TC. The general ATM Insert/Extract Block as

described in **Chapter 4.2.6** is also available in this application. Interworking scenarios shown in the CPE section will also be supported in CO applications.

4.2.9 CO EFM Application

When using the SOCRATES™-4e on the CO side (STU-C) in EFM application, only PHY mode is supported (see **Chapter 1.3.3.2**). The used blocks are identical to the CPE EFM PHY applications (see **Chapter 4.2.2**), therefore the description given in that chapter applies to the CO application as well. The PFA block supports up to 4 groups.

4.2.10 CO Packet Application

As in CO EFM applications, only PHY mod is supported (see **Chapter 1.3.3.2**). The used blocks are identical to the CPE Packet Application in PHY mode (see **Chapter 4.2.4**), this chapter also covers a functional description of the used blocks. The SOCRATES™-4e offers in CO Packet Applications the same bonding capabilities as in CO EFM applications (see **Table 1**).

4.2.11 Dual Bearer Applications

All the above application can also work in the so called dual bearer mode. In this case the bandwidth of the SHDSL line will be spit in the 2 parts, the first bearer is always the TDM bearer, the second one out of the above described applications. Note that IEEE 802.3-2004 does not support dual bearer application.

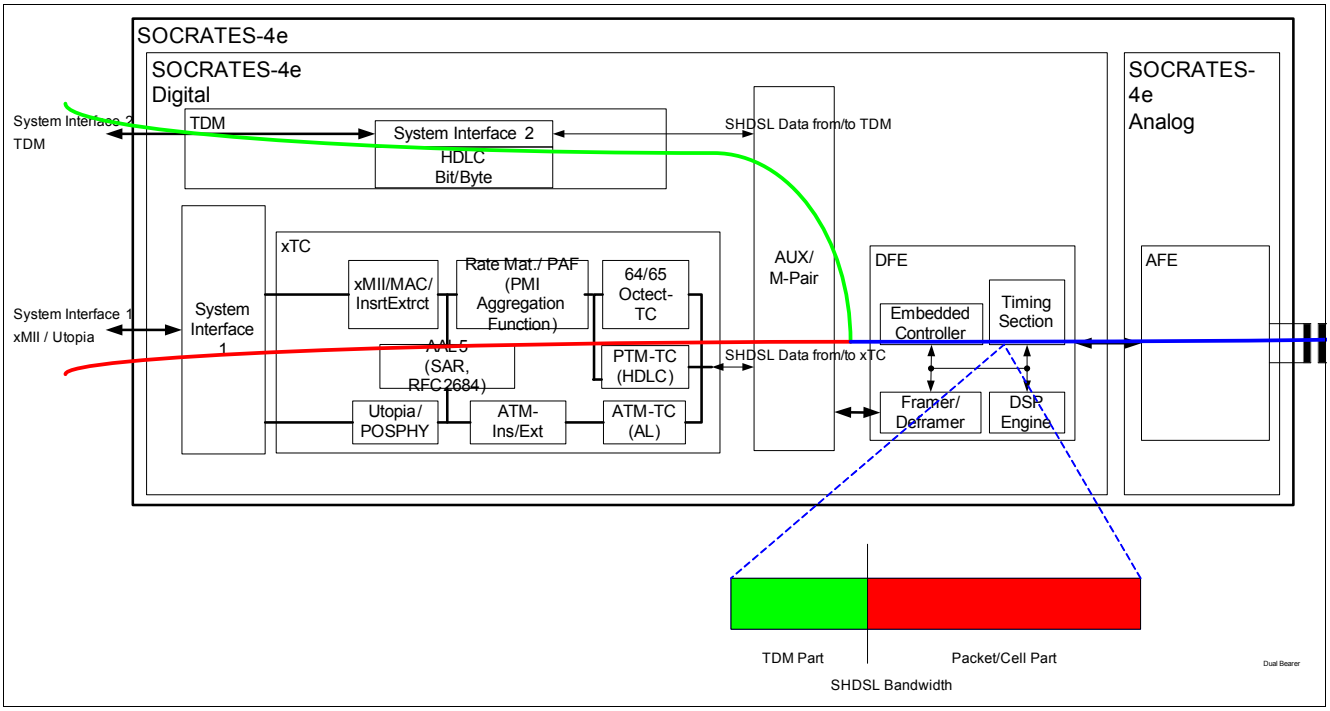


Figure 52 Dual Bearer Application

4.2.11.1 TDM Block

The TDM block is directly connected to the TDM interface (System Interface 2). This block realizes all TPS-TCs standardized in ITU-T G.991.2 (2004) except the packet or cell based ones which are handled by the xTC.

The TPS-TCs handled by the TDM are referred to as Serial TC in this document.

The TDM block can be connected to all four SHDSL lines at the same time if these SHDSL lines work as independent links. In this mode the TDM interface has to work in bus mode with dedicated time slots for each SHDSL link.

Up to 8 consecutive bits of one SHDSL link can be extracted and inserted by the IDC via the HDLC controller. The HDLC controller can be configured as bit or byte oriented and only works towards the SHDSL line. The HDLC controller can be switched on-the-fly from one SHDSL link to another. The SHDSL links for which HDLC controlling is enabled and which are not connected to the HDLC controller are transmitting idle flags in the appropriate bits of the sub-payload blocks.

4.2.12 General Blocks

This chapter describes general blocks which are not specific to 1 application

4.2.12.1 MDIO

The MDIO holds the register set according clause 45 of IEEE 802.3-2004.

Clause 22 register set according to IEEE 802.3-2004 is accessible by the MDIO.

In PHY mode the register set according to clause 45 is exchanged with the external MAC via the maintenance data interface. The register set according to clause 22 does not exist in this mode.

In MAC mode the register set according to clause 22 is exchanged with the external Ethernet PHY via the maintenance data interface. The register set according to clause 45 is exchanged with the IDC towards the PMD.

4.2.12.2 M-Pair

According to ITU-T G.991.2 (2004) up to 4 SHDSL wire pairs can be aggregated to a M-pair bundle. This bundling technique is realized by the M-pair block. The differential delay between the SHDSL wires forming an m-pair bundle will be measured in this block, up to 125 µs differential delay (beyond standard) can be internally compensated, i.e. no external buffer is required.

4.2.13 IDC

The IDC (Integrated Device Controller) controls and maintains all building blocks of the SOCRATES™-4e. The Host controller has access to the IDC via the microprocessor interface to configure the SOCRATES™-4e, to read out the status and to handle the inserted and extracted data from the xTC. The following subclauses describe the OAM handling of the xTC.

4.2.13.1 OAM for Ethernet (Towards System Interface 1)

Ethernet OAM functionality is provided in MAC mode.

The IDC provides the counters required for DTE management (basic and mandatory package) according to clause 30 of IEEE 802.3-2004. MIB objects defined in RFC2665/RFC3635 and RFC2863 are supported as well.

An externally connected Ethernet PHY is controlled by the IDC via MDIO according to clause 22.

4.2.13.2 OAM for EFM (Towards SHDSL Link)

In PHY mode the IDC provides the registers of each SHDSL line for an external MAC according to clause 45 of IEEE 802.3-2004. In MAC mode these registers are accessible by the external Host via the microprocessor interface.

In MAC mode also the counters required for DTE management (basic and mandatory package) according to clause 30 of IEEE 802.3-2004 for the SHDSL link are provided, EFM OAM as defined in clause 57 is supported and the MIB objects defined in RFC2665/RFC3635 and RFC2863 are supported.

4.2.13.3 OAM for ATM

F4/F5 OAM (AIS, RDI, CC, LB) according to ITU I.610 is fully supported by the IDC. The appropriate cells are extracted/inserted in the ATM Insert/Extract block of the xTC.

4.2.14 Digital Front End

The Digital Front End of the SHDSL transceiver is in compliance with ITU-T G.991.2 (2004) and with the digital part of the PMD according to IEEE 802.3-2004.

The building blocks are described in more detail in the following subclauses.

4.2.14.1 DFE RAM

Firmware for the 4 SHDSL channels will be stored within this integrated memory block. Firmware download is done via the IDC.

4.2.14.2 Timing Section (Timing Recovery)

On the central office side (STU-C) the loop clock can either be derived directly from the local oscillator or can be synchronized to the network reference clock or transmit data clock by means of a digital phase locked loop.

The network timing reference on the remote side (STU-R) is recovered from the received frame clock by means of a digital phase locked loop. This network timing reference is supplied as reference clock to the system for all data rates.

If for an application the range of the payload data rate is limited from 192 kbit/s to 5696 kbit/s per link, which is according to the ITU-T G.991.2 standard, one crystal of 20.48 MHz can be applied to minimize power dissipation or to be hardware compatible to Lantiq's SDFE-4 family.

Only for payload data rates above 5696 kbit/s one crystal or oscillator of 40.96 MHz has to be applied for optimum performance results.

4.2.14.3 Framer and Deframer

In transmit direction the framer puts the payload and the overhead bits in the defined order of a SHDSL frame. In receive direction the deframer extracts the payload and overhead bits out of the frame.

4.2.14.4 DSP Engine

The DSP engine consists of various sub-blocks which are described in this chapter. Both 16 and 32 level PAM are supported, furthermore the SOCRATES™-4e can also be operated with 4, 8, 64 and 128 level PAM.

Trellis Encoding

The trellis encoder performs signal space coding based on a finite state machine (convolutional encoder). The rate 1/2 trellis encoder introduces redundancy and dependence of the successive symbols by its state memory. Coding gain is achieved by convolutional encoding and due to constellation design.

Viterbi Decoding

Viterbi decoders allow optimum decoding of convolutional codes in the receiver. The Viterbi algorithm is a sequential trellis search algorithm for performing maximum likelihood sequence detection.

Echo Cancellation

Echo cancellation enables bidirectional data transmission using the same frequency band. An echo is an undesired interference of the transmit data signal into the receiver through the hybrid. The adaptive echo canceller generates a replica of the echo which is subtracted from the receiver input to yield an echo-free signal.

Tomlinson Precoding

The trellis encoding and Viterbi decoding does not allow the use of a decision feedback equalizer in the receive path. Transmitter precoding (Tomlinson-Harashima coding) achieves a performance essentially identical to the decision feedback equalization by moving the feedback filter to the transmitter.

Scrambling/Descrambling

A maximum-length shift register is used on the input bit stream to “randomize” or “whiten” the statistics of the data, making it look more random. Scrambling of the transmitted signal is reversed by a descrambler at the receiver.

Decimation and Filtering

Decimation reduces the data rate of the highly oversampled bit stream provided from the Sigma-Delta Analog to Digital Converter. All of the out of band noise is eliminated within the filter stages.

Interpolation and Noise Shaping

The data rate is increased to get the oversampled data which is necessary for the digital to analog conversion.

HDLC Controller for EOC

This HDLC controller (one per SHDSL channel) handles EOC messages in an HDLC-like format as defined in ITU-T G.997.1 paragraph 6.2 and standardized in ITU-T G.991.2, ITU-T G.991.2 (2004) and ETSI TS 101 524.

It performs the following framing functions: flag generation/recognition, byte stuffing and CRC check.

One FIFO for the receive and one for the transmit direction are implemented to allow EOC messages of every length to be transported within one HDLC frame.

4.2.14.5 On-Chip Biterror Tester

An on-chip biterror tester is available for each channel.

In the biterror test mode scrambled ones or zeros are transmitted in the payload data stream.

The receiver counts the received zeros and ones of the payload data as failed bits within a counter period of n*6 ms (BERT Counter), where n is any programmable integer value between 1 and 65536.

The counter stops if its maximum value is reached within one counter period.

For host access the BERT Counter value is stored in the BERT register at the end of the counter period and the BERT Counter is restarted.

While BERT is enabled ’1’ are transmitted to the Serial Data Interface.

The principle of the on-chip biterror tester is shown in **Figure 53**.

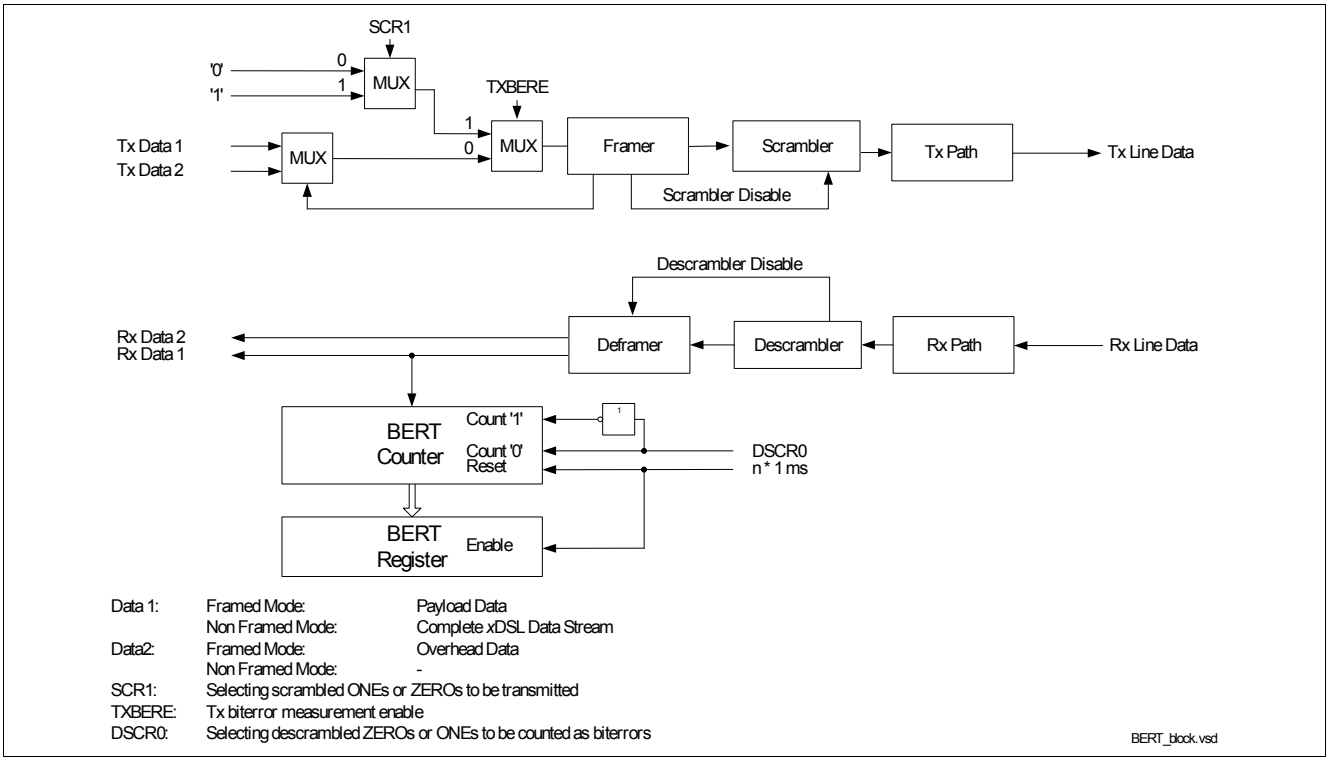


Figure 53 Block Diagram, Biterror Test

4.2.14.6 Embedded Controller

For preactivation (G.hs), activation/deactivation and the HDLC handling of the EOC channel one controller for each SHDSL channel is integrated.

Before activation the firmware for the microcontroller has to be downloaded via the IDC to the integrated DFE RAM.

4.2.14.7 Test Modes

The DFE supports various test modes.

- Test and Audible Tone - The DFE transmits a tone at a programmable frequency (800 Hz - 300 kHz). Based on that tone i.e. a line identification in a cable bundle is possible
- PSD Measurement Mode
- TDR Measurement
- Idle Noise Measurement (INM)

4.3 Analog Front End

The Analog Front End is in compliance with ITU-T G.991.2 (2004).

The building blocks are described in more detail in the following subclauses.

4.3.1 Block Diagram of the AFE

Beside the common 20.48/40.96 MHz oscillator clock the AFE is identical for each channel (see **Figure 54** and **Figure 55**). For channel 0 an oscillator is implemented in addition. The oscillator clock of channel 0 is supplied to all other channels and to pin CLKOUT (see **Figure 56**).

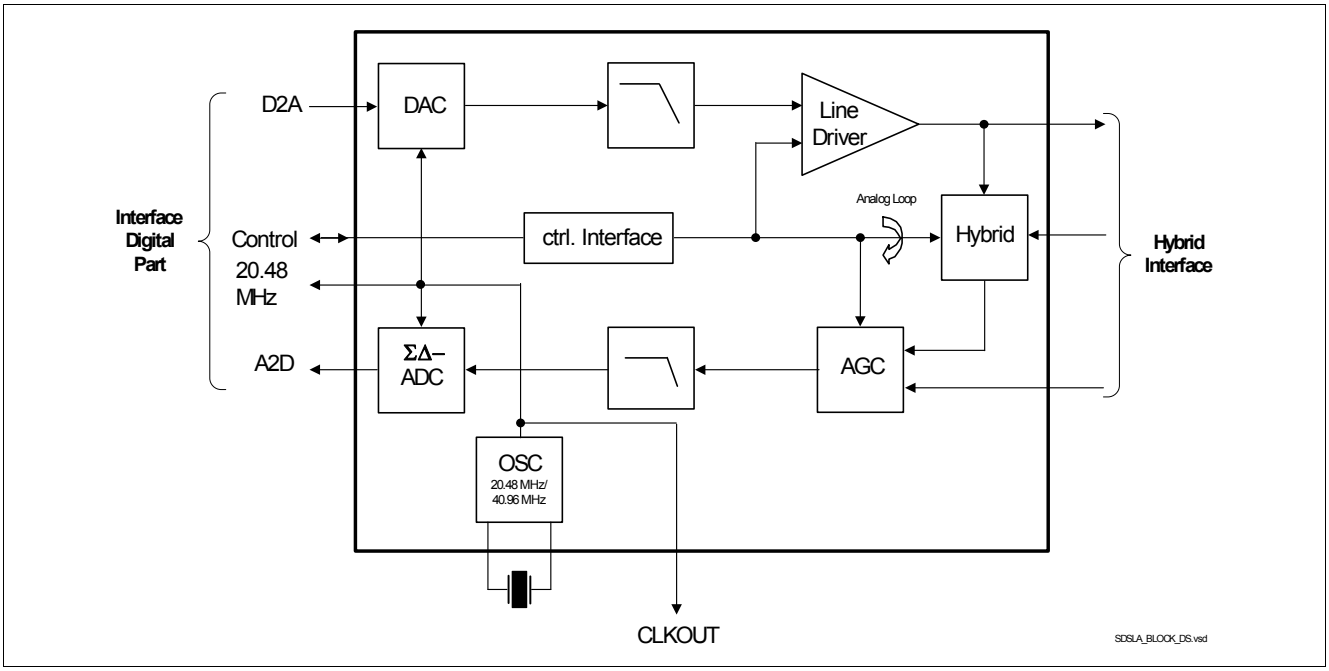


Figure 54 Block Diagram of the AFE, Channel 0

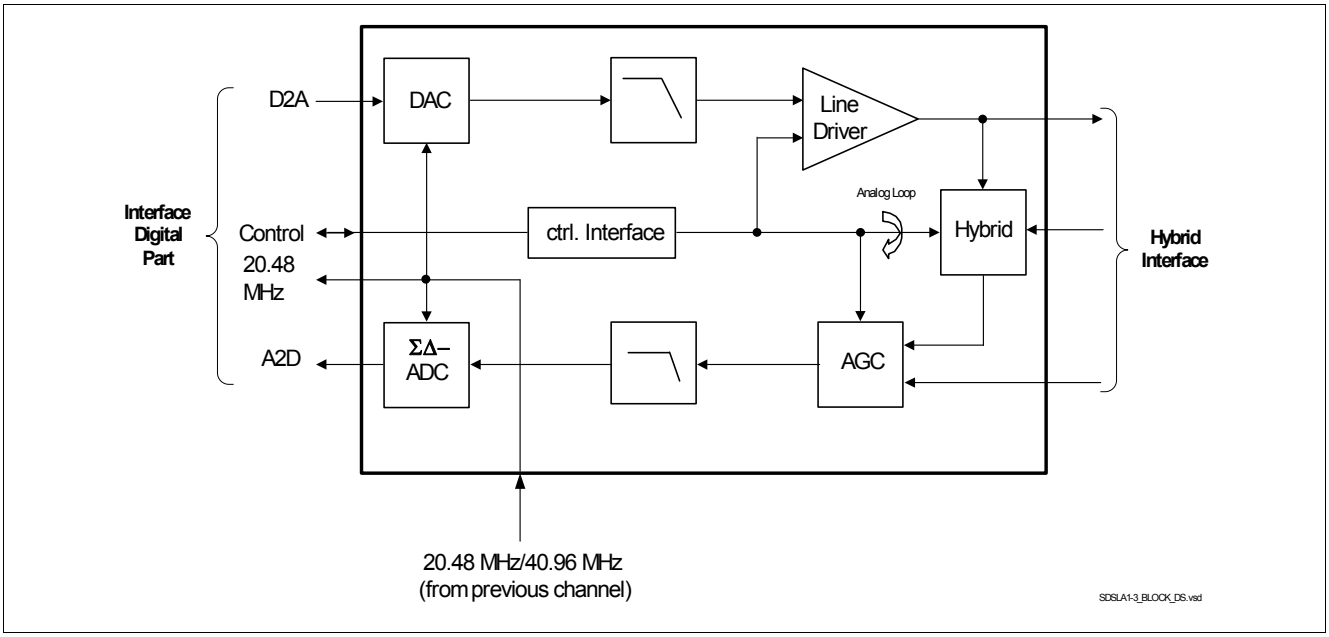


Figure 55 Block Diagram of the AFE, Channel 1, 2 and 3

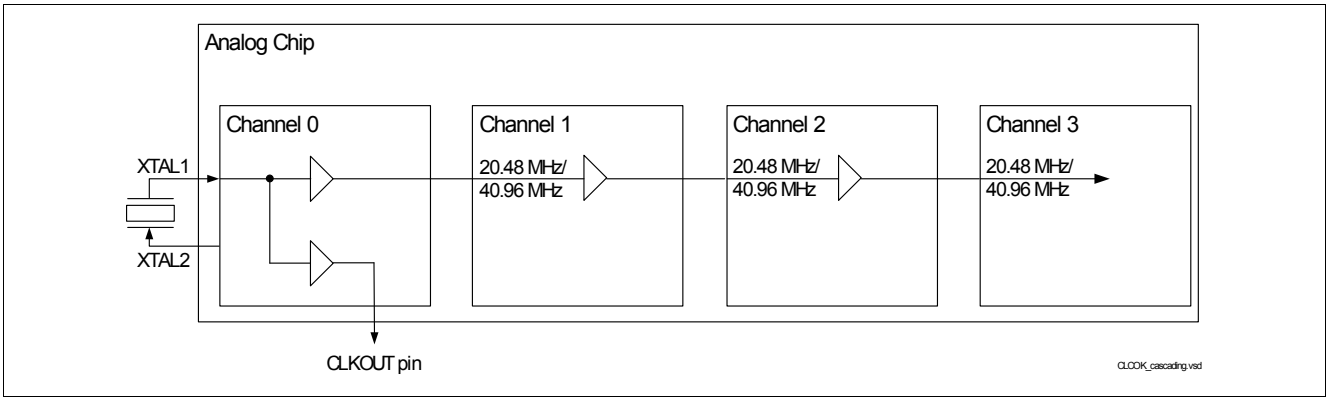


Figure 56 Clock Cascading of the Oscillator Clock

4.3.2 Features

The integrated analog front end is designed for SHDSL applications with data rates up to 2560 kBaud/s. All symmetrical and asymmetrical PSDs are supported according to ITU-T G.991.2, ITU-T G.991.2 (2004) and ETSI TS 101 524.

It integrates all receive and transmit functional blocks:

- Automatic Gain Control, AGC
- Filtering
- Analog to Digital Conversion, ADC
- Digital to Analog Conversion, DAC
- Crystal Oscillator with frequency tolerance of ± 60 ppm over temperature and aging[1)]
  - For payload data rates from 192 kbit/s to 5696 kbit/s: XOf = 20.48 MHz
  - For payload data rates above 5696 kbit/s: XOf = 40.96 MHz
- Hybrid, automatically tuned to the real line conditions
- Line driver

4.3.3 Digital/Analog Converter DAC

The digital symbols are transformed to an analog symbol via a low power multi bit current steering DAC, which has high resolution and is highly linear in order to maximize the echo cancellation. A differential type of line driver with a high current driving capability and low distortion characteristics drives the subscriber lines.

4.3.4 Line Driver

A differential type of line driver with a high current driving capability and low distortion characteristics drives the subscriber lines.

4.3.5 Analog/Digital Converter ADC

The analog input from the Automatic Gain Control amplifier is converted into digital data by a third order sigma-delta modulator. Depending on the oscillator frequency the converter's sampling rate is either 20.48 MHz or 40.96 MHz.

Oversampling reduces the amount of quantization noise power present in the signal band. Noise shaping further attenuates quantization noise in the signal band, thereby pushing noise power to out-of-band frequencies. The noise power, that is pushed outside the signal band can be attenuated by a digital filter such that it has no further effect on the signal.

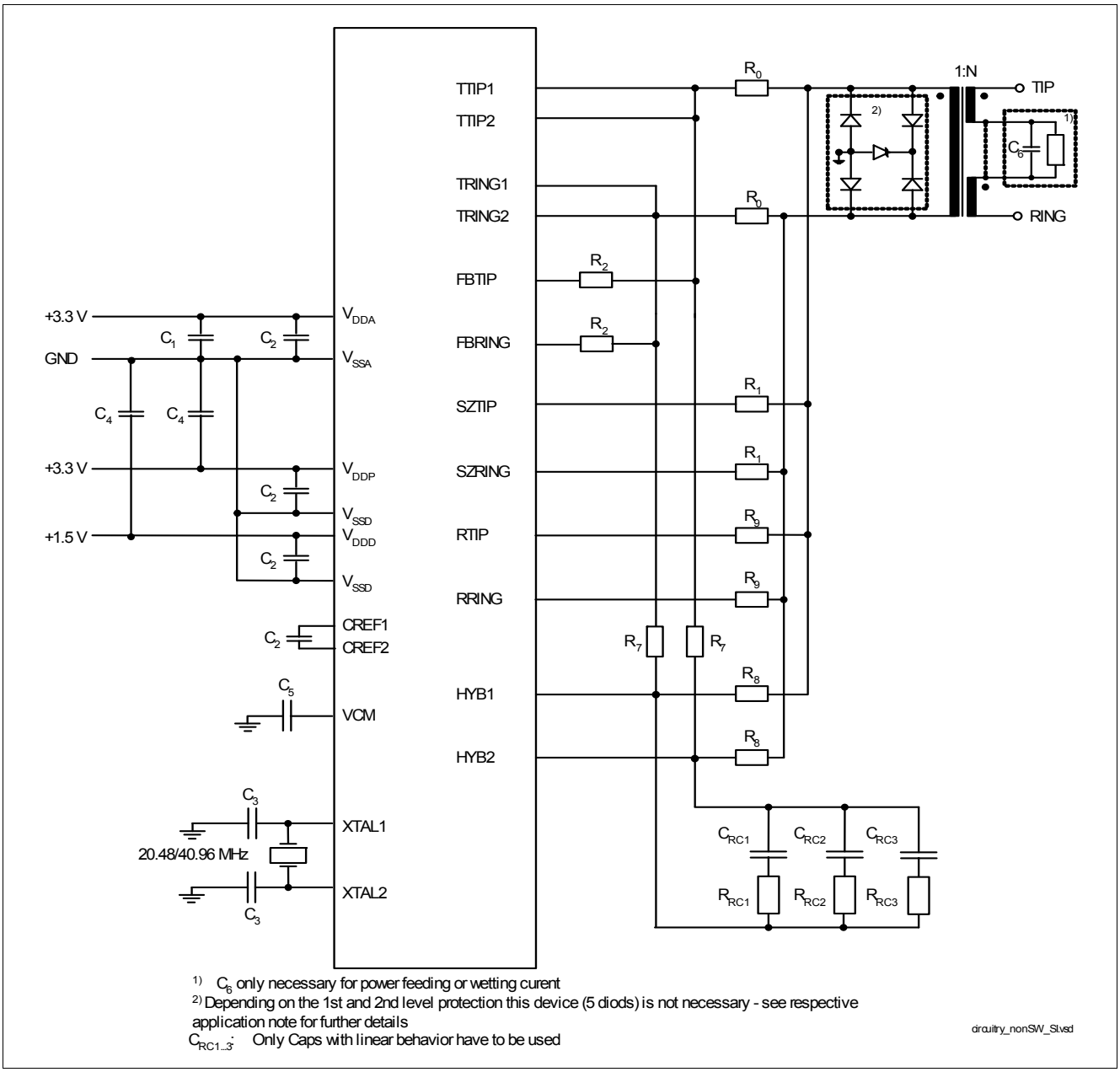
1) For a detailed specification of the crystal refer to **Chapter 7.7**

4.3.6 Analog Echocanceller (HYBRID)

The dominant disturber in duplex data transmission over twisted-pair cables is the echo leaking from the transmitter into the receiver. Therefore a B-Filter (Balancing Filter, adaptive hybrid) is implemented in each analog part. Adaptivity is essential as the input impedance of the loop depends on the wire gauge, gauge changes and the presence/absence of bridged taps. The adaptation algorithm is handled by the data pump at system startup.

4.3.7 External Circuitry (for Analog Part and Power Supply)

The external circuitry described in this chapter applies to symmetrical PSDs. For asymmetric PSDs an optimized external circuitry compared to this chapter is required.



1) C_6 only necessary for power feeding or wetting curent
2) Depending on the 1st and 2nd level protection this device (5 diods) is not necessary - see respective application note for further details
$C_{RC1...3}$: Only Caps with linear behavior have to be used

Figure 57 External Circuitry for one Channel

Note: Only passive components are required

The values of **Table 30** are recommended for all applications using symmetrical PSDs.

Table 30 External Circuitry

Symbol	Value[1)]	Unit	Tolerance
N	4.5		See Chapter 7.8
R0	2.3	Ω	± 1 %
R1	43	Ω	± 1 %
R2	43	Ω	± 1 %
R7	750	Ω	± 1 %
R8	3.3	kΩ	± 1 %
R9	1.3	kΩ	± 1 %
RRC1	2.2	kΩ	± 1 %
RRC2	5.6	kΩ	± 1 %
RRC3	12.7	kΩ	± 1 %
C1	220[2)]	µF	± 20 %
C2	100	nF	± 20 %
C3[3)]	6.8	pF	± 5 %
C4	4.7	µF	± 20 %
C5	100	nF	± 10 %
C6	1	µF	± 10 %
CRC1	3.3	nF	± 5 %
CRC2	1	nF	± 5 %
CRC3	470	pF	± 5 %

1) CRC1...3: Only capacitors with linear behavior (linear dielectric material - COG or NP0) have to be used

2) Recommended size, smaller values are possible

3) This capacity includes the board capacity, i.e. C3=2*CL-Cboard. Therefore this capacity value depends on the board design and has to be adapted accordingly

A value of 1 µF is recommended for Capacitor C6 to guarantee full performance also for low data rates. If smaller capacitors are required due to layout considerations, this might have an impact, mainly on lower data rates. Tests have shown that for example a capacitor of C6 = .33 µF and C6=0.47 µF show a performance reduction (SNR) of 1-1.5 dB at 192 kbit/s

5 Interface Description

The following subclauses specify the functional timings of the interfaces of the SOCRATES™-4e.

5.1 Control Interface

The SOCRATES™-4e offers 3 different interface for controlling the chip. The supported controller interface are the microprocessor interface, the RS232 and Serial/Parallel Interface and the Serial Control Interface (SCI). Note that the availability of an interface depends on the application (see **Chapter 1.3.3**) and will be controlled using the pins CCONF(2:0) (see **Chapter 2.2.2**).

The SOCRATES™-4e will be configured using messages, for further details refer to the User's Manual - Programmer's Reference. These message will be used in all different control interface application.

5.1.1 Microprocessor Interface

In the following, the Microprocessor Interface is abbreviated as MPI. The MPI supports direct access via virtual register to the internal Device Control Interface (DCI) via a 8-bit wide data bus. The DCI area is 32-bit structured and described in the User's Manual - Programmer's Reference.

Features

- Acts as a slave in the external host system
- Supports Intel Demux Mode (little endian) CCONF(2:0) = 000_B
- Supports synchronous Motorola Mode (big endian) CCONF(2:0) = 010_B
- Supports asynchronous Motorola Mode (big endian) CCONF(2:0) = 001_B
- 8-bit data bus width
- Ready/DataTransferAcknowledge controlled bus[1)]
- Optimized for fast access and minimum waitstates
- 128 byte FIFO (per direction) for message exchange
- Connected to 8K x 32 bit Device Control Interface (DCI) area for message exchange and virtual registers
- All RAM cells are cleared automatically upon reset

External interrupt generation to support handling of message queues and virtual registers.

The functional timing for the microprocessor interface is according to **Chapter 7.11.1**.

5.1.2 RS-232 Interface and SPI Interface

The RS-232 and the SPI interface will always be selected together by setting the CCONF(2:0) = 011_B. This interface will typically used in so called stand-alone applications, no additional controller is available for controlling the SOCRATES™-4e. It is expected that the entire firmware and the configuration of the chip (number of messages in certain order) is stored in an external flash memory and the contents will be retrieved after start up of the device. The RS-232 interface might be used for monitoring and debugging purposes. Please refer to FW release notes if FW package for stand alone mode is supported.

5.1.2.1 Serial Peripheral Interface

The Serial Peripheral Interface (SPI) is used to access the external Flash. The SOCRATES™-4e supports the flash types of the following supplier:

- ST
- SST

A minimum flash memory size of 8 Mbit is required for storing the firmware and the configuration information. Therefore the following types are recommended:

1) Connection of Ready/Data TransferAcknowledge signal not necessary if required timing can be fulfilled by other means

- M25P80, Code Flash, 8 Mbit, ST
- SST25VF080, 8 Mbit, SST

The following figures illustrate accesses to the flash memories. Refer to **Chapter 7.11.1.5** for timing characteristics.

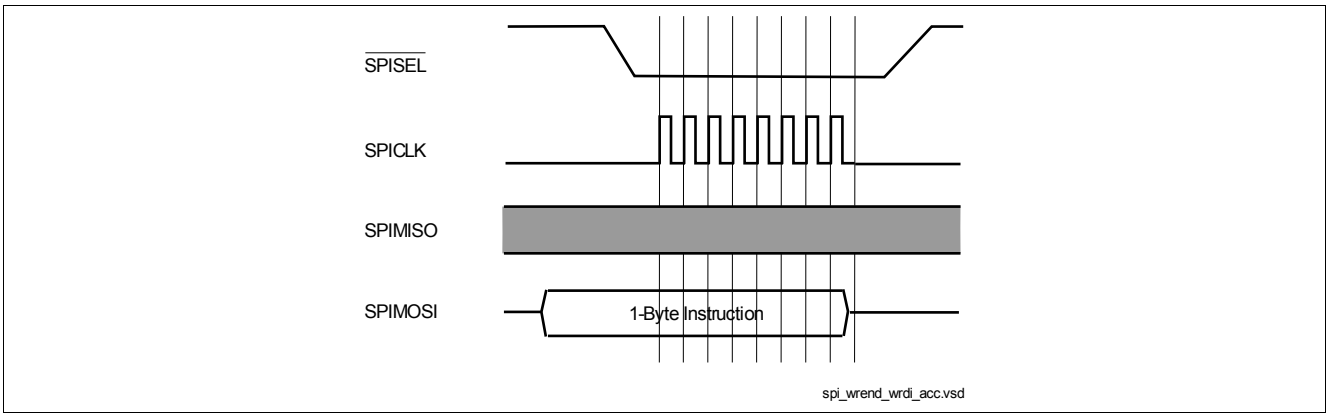


Figure 58 SPI Write Enable/Disable (WREN/WRDI)

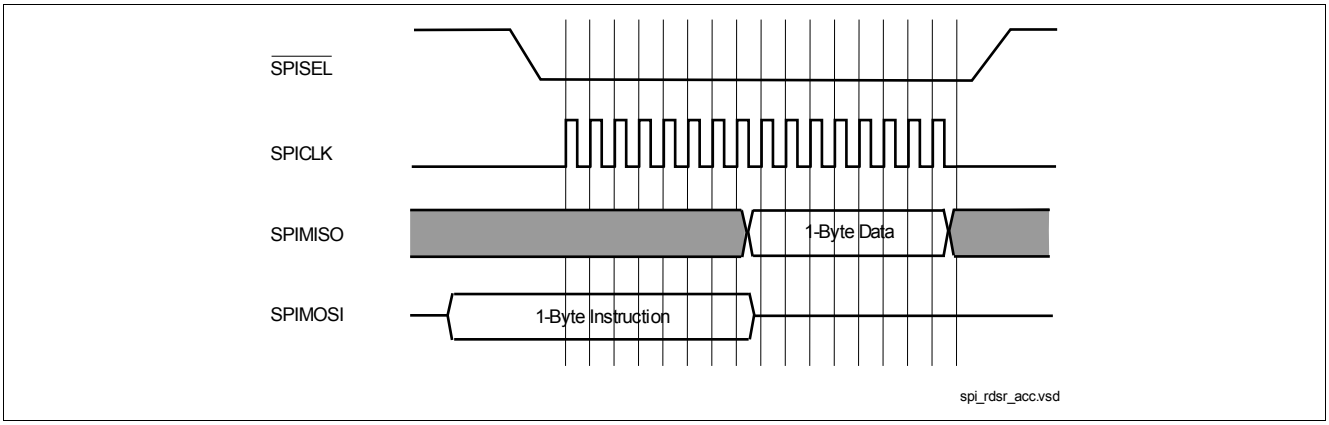


Figure 59 SPI Read Status Register (RDSR)

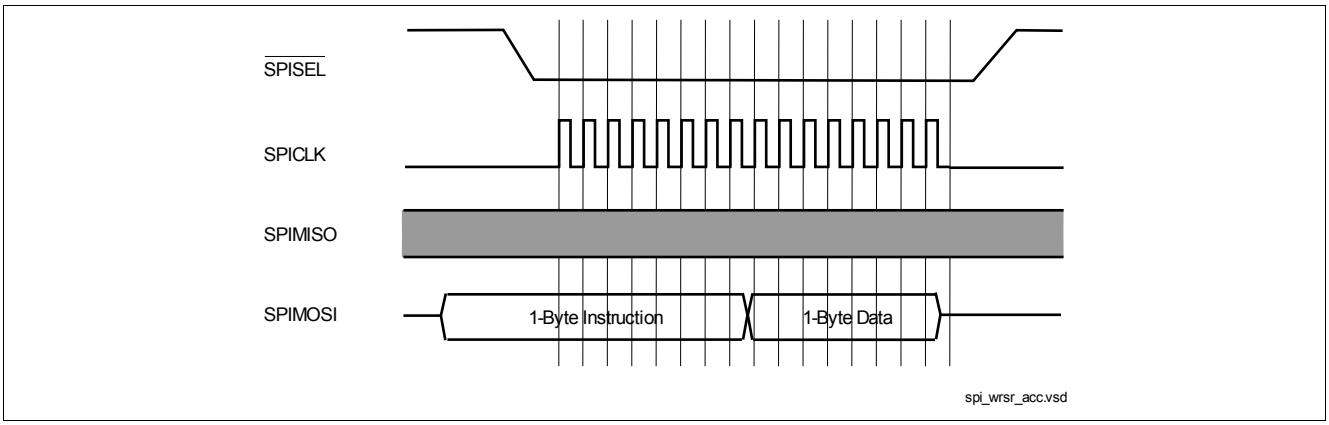


Figure 60 SPI Write Status Register (WRSR)

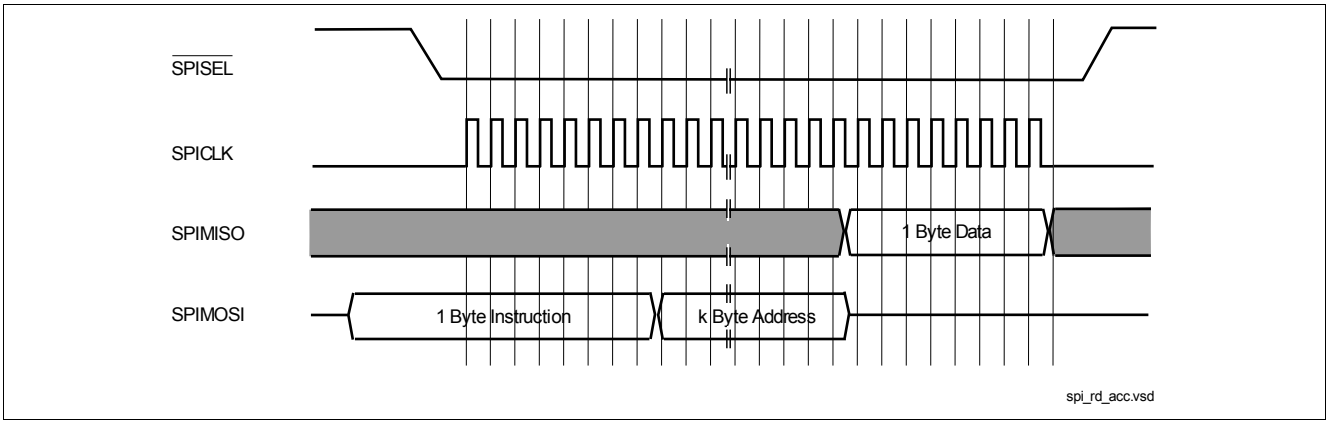


Figure 61 SPI Read

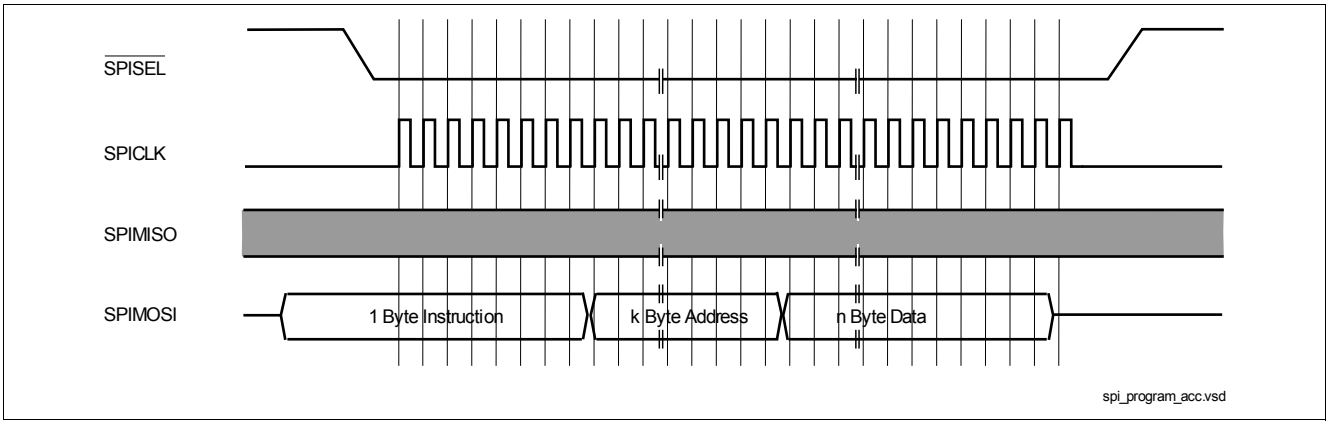


Figure 62 SPI Program

5.1.2.2 RS-232 Interface

The RS-232 interface is based on a 11 bit UART frame with eight data bits (LSB first), one stop bit, one start bit and one parity bit

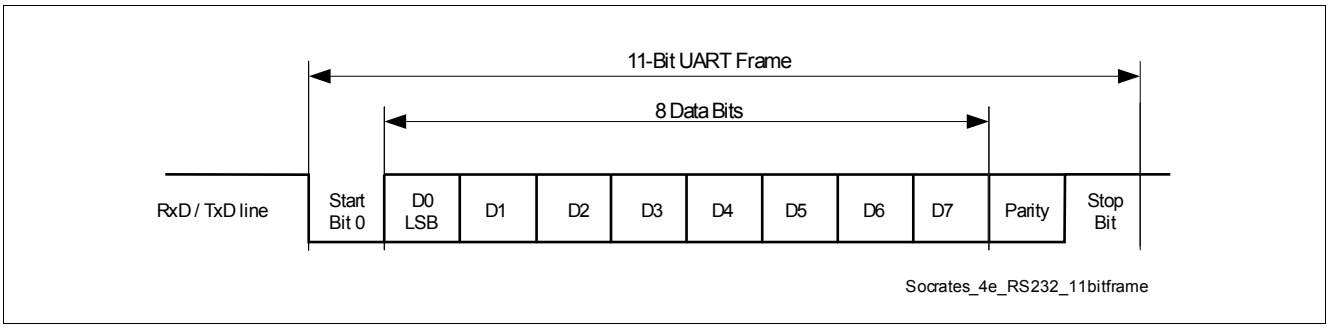


Figure 63 RS-232 (UART) Frame

5.1.3 Serial Control Interface

The Serial Control Interface (SCI) will be selected by setting CCONF(2:0) = 100_B. Control and configuration information will be exchanged HDLC framed. Therefore a 128 byte message FIFO is available per direction.

5.1.3.1 Collision Handling

Up to 15 SOCRATES™-4e can be connected via 1 SCI interface. The following figure shows a system where 4 SOCRATES™-4e are connected to an external host via the SCI interface. **Figure 64** shows the use of the DADDR pins which is explained in **Chapter 5.1.3.3**.

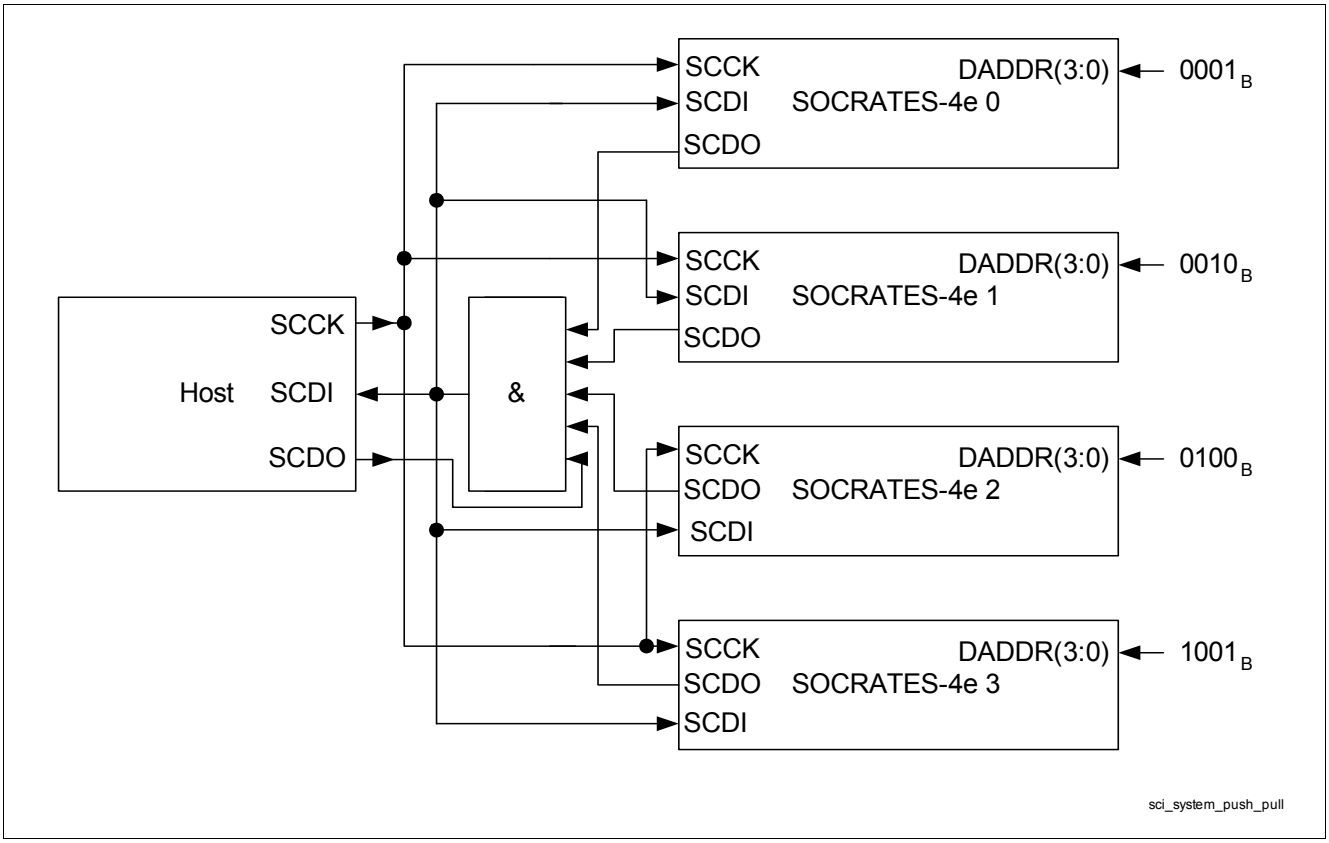


Figure 64 SCI System

Since all the different SHDSL line can generate notification on arbitrary times (not correlated to notification of other channels) the SCI interface has implemented a collision handling function. Furthermore, in some cases the internal controllers have to communicate with each other to exchange messages, for example in the M-pair mode. This communication beside the communication with the external host controller is performed via the external SCI. To avoid collisions the SCDI pin of the external SCI is combined by an AND with all outputs of the internal modules. This combined signal is transmitted via pin SCDO and ANDed with the SCDO of the other SOCRATES™-4e and fed back to the internal modules. The modules transmitting a logical ’1’ but receiving a logical ’0’ will stop transmission and will retry transmission after the next HDLC Stop Flag.

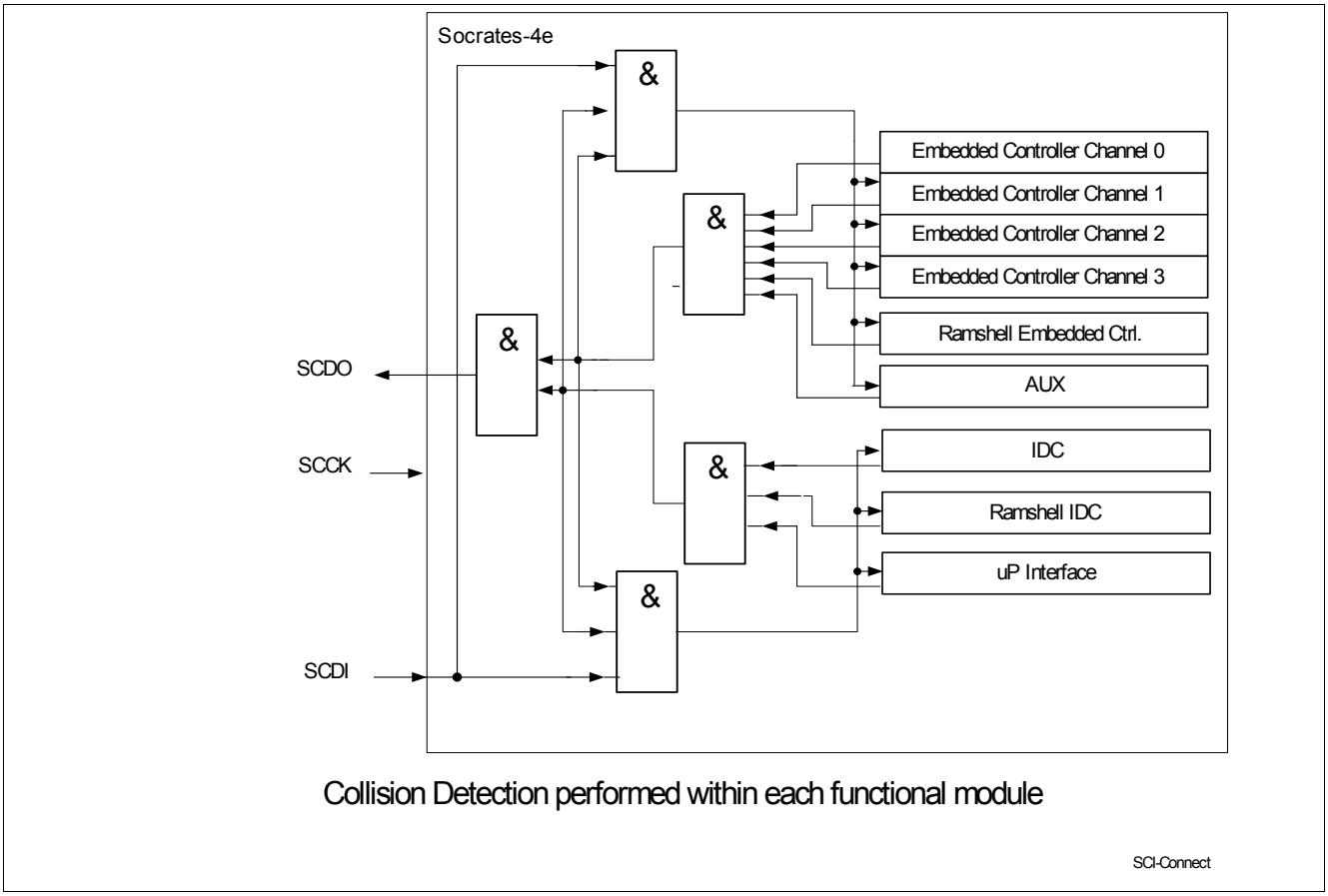


Figure 65 Internal Connection of SCI Interface

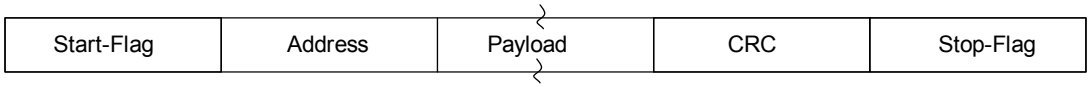
This communication protocol is based on LAPD and according to ITU-T Recommendation I.430. Based on this protocol only 1 sender can send data. In case of simultaneous start of transmission of 2 sender, the frame with the first ’0’ wins, the other has to stop sending. As long as 1 sender transmits data, the other connected sender may not send data.

To ensure the message exchange between the host and the SOCRATES™-4e the mechanism of collision detection and withdrawal from transmission in case of a collision has to be realized in the host controller as well. For collision detection the host controller has to use pin SCDO not only as the message direction from SOCRATES™-4e to host but also as echo of the message transmitted by the host.

The SCI interface is available in all different interface modes. For debugging purposes it is recommended to connect this pins to test ports.

5.1.3.2 SCI Protocol

The transport protocol of the SCI interface is HDLC and according to ITU-T Recommendation Q.921 (HDLC bit oriented). To avoid 6 consecutive ONEs a ZERO is transmitted after 5 consecutive ONEs. The header and payload bytes are transferred LSB first whereas the CRC is transferred MSB first. The HDLC frame is shown in **Figure 66**.



Start/Stop-Flag: HDLC Flag ($7E_H$)
Address: Contains Source and Destination Address (2 Bytes)
Payload: User Payload (Message)
CRC: CRC sum over entire message (16 bit)

HDLC Frame

Figure 66 HDLC Frame

After transmission of the stop-flag continuous ONEs (at least nine ONEs) are used as inter-frame fill.

To decouple the SCI performance from the performance of the embedded controllers one 128 byte FIFO is implemented for the receive and one for the transmit direction for each controller. The FIFOs are implemented as cyclic buffers.

The AUX and RAM blocks are directly accessed, no FIFO's are needed for these modules.

5.1.3.3 Address Mapping

The address field of HDLC frame consists of 2 bytes and is described in **Table 31**. The entire received address field of the incoming HDLC frames is compared. Just frames with DESTADDR which match the values of the module address are accepted.

Table 31 Address Field

High Address Byte	Low Address Byte
SOURCEADDR	DESTADDR

SOURCEADDR Source Address (Address of Transmitter)
DESTADDR Destination Address (Address of Receiver

Each functional block of the SOCRATES™-4e has its own address comprising the device address and the assigned block address. Each part consists of 4 bit, the device address maps to bit 7-4.

The 4 bit device address (unique for 1 SOCRATES™-4e) allows the connection of up to 15 SOCRATES™-4e to one SCI. The device address F_H is reserved and is used as a broadcast address. The device address is latched in from pins DADDR_0, DADDR_1, DADDR_2 and DADDR_3. (DADDR_3 will be mapped to bit 7 of the address field, DADDR_0 t0 bit 4 - see **Table 32**). A module within a chip is addressed correctly if the device address part either contains the device address specified by the DADDR pins or the broadcast address (F_H).

Table 32 Device Address

PIN	Device Address Bit	Bit in Address Field
DADDR_3	3	7
DADDR_2	2	6
DADDR_1	1	5
DADDR_0	0	4

The following table list the block addresses of the internal modules:

Table 33 Block Address

Block	Address(3:0)
IDC	3
Ramshell IDC	8
uC Interface	4
Ramshell Embedded Controller	5
AUX	9
Embedded Controller Channel 0	2
Embedded Controller Channel 1	6
Embedded Controller Channel 2	A
Embedded Controller Channel 3	E

When the SOCRATES™-4e sends a message to the external controller it uses $F3_H$ as destination address.

5.2 System Interface 1

This chapter describes the different modi of the System Interface 1. This interface can be operated in xMII mode or in UTOPIA/POS-PHY mode. When working in xMII mode, MAC and PHY mode are supported in general. Details of the supported modes can be found in the following chapters.

5.2.1 xMII Interface

The xMII interface is one mode of the System Interface 1 and will be used for payload data transfer towards the system. It is expected that Ethernet frames according to IEEE 802.3-2004 are carried over this interface.

The following 2 figures define the direction Receive and Transmit at the xMII interface in the different operating mode.

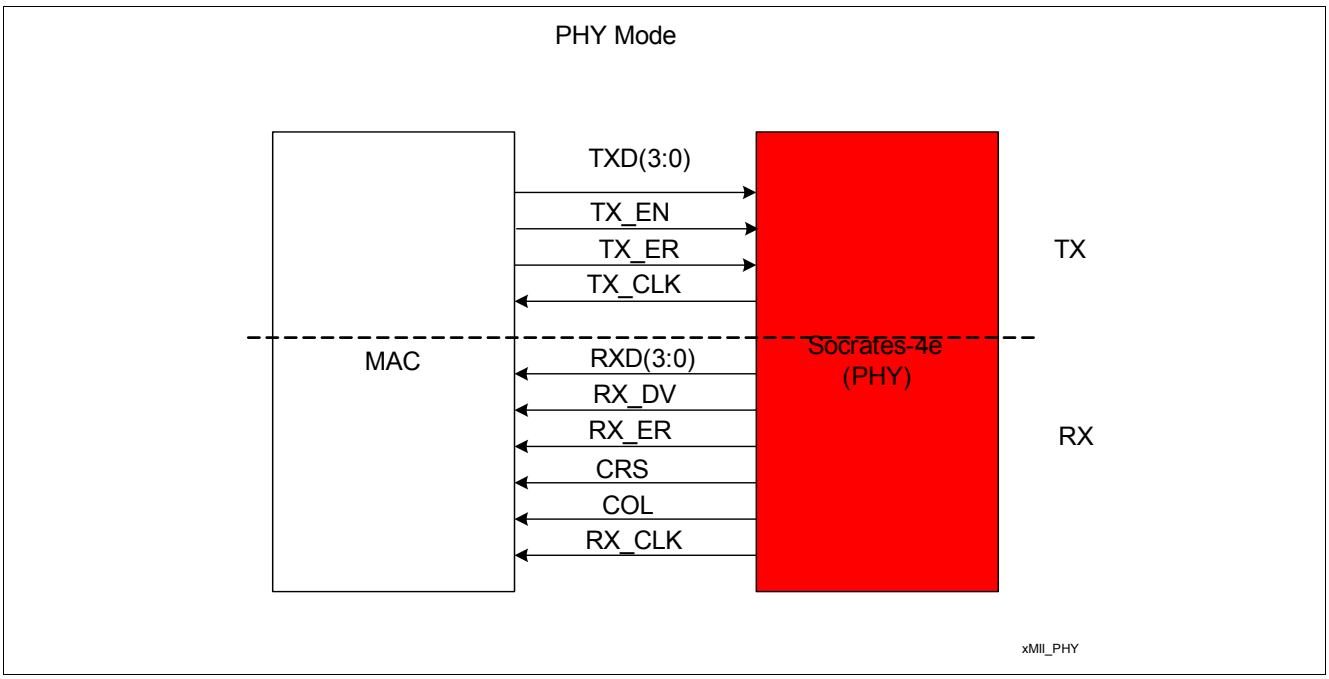


Figure 67 MII Interface PHY Mode

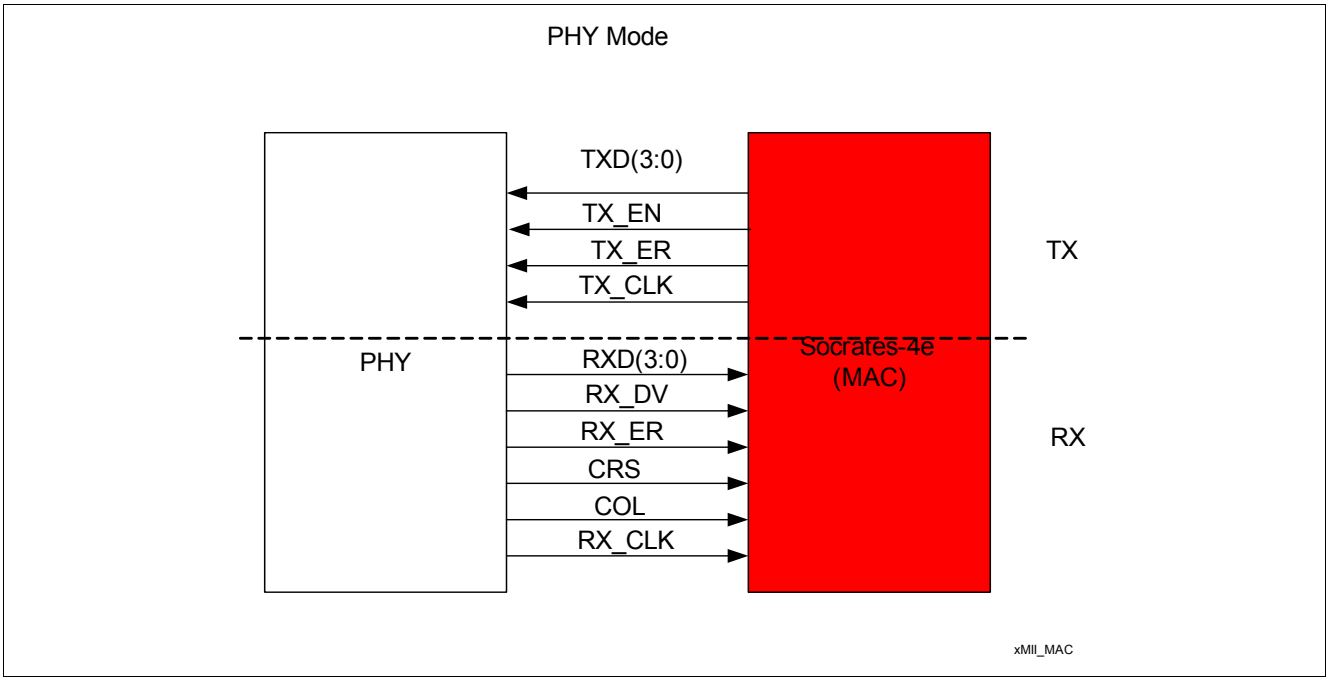


Figure 68 MII MAC Mode

From a SOCRATES™-4e perspective, the transmit direction signals are defined as inputs and the receive direction signals are defined as outputs in PHY mode (see **Figure 67**), whereas in MAC mode transmit direction signals are defined as outputs and the received direction signals as inputs (see **Figure 68**).

In general 3 different flavors of the xMII interface are supported (in MAC mode only MII and SS-SMII interface supported):

- MII interface
- RMII interface
-
- SS-SMII interface

The particular mode will be configured using the pins TCCONF(2:0). The following table lists the number of interface in the different MII configurations.

Table 34 Number of xMII Interface

xMII Interface Configuration	Number of Interface	Default Mapping	Support of PAF
MII	1	SHDSL channel 0	yes
RMII	4	1: 1 mapping, (RMII channel 1 to SHDSL channel 1, ..	yes[1)]
SS-SMII	4	1: 1	yes[1)]

1) IF PAF is used, not all xMII interfaces are used (i.e. SHDSL channel 0 and 1 form a PAF group and SHDSL channel 2 and 3 do not use PAF, only 3 xMII interfaces used)

Loop Back

Loop backs are supported in all different xMII flavors. All the loops are non-transparent. 2 loops are available, the ingress loop mirrors the data received at the xMII interface back to the output. The egress loop mirrors data to be sent out at the xMII interface back to the internal blocks of the SOCRATES™-4e.

These loops are available per xMII interface (see **Table 34**) and can be programmed independently (direction of loop) per port.

5.2.1.1 xMII Interface in PHY mode

In PHY mode all different MII flavors are available. It is expected that this interface is operated in half duplex mode. **Table 11** shows the available pins together with their direction.

MII Mode

The MII interface is implemented according to IEEE 802.3-2004. For rate matching (see **Chapter 4.2.2.1**) the signal CRS will be used. The SOCRATES™-4e provides the TX_CLK and RX_CLK of f = 25 MHz.

RMII Mode

The RMII interface is implemented according to RMII specification of the RMII consortium. It is expected that the REF_CLK (f = 50 MHz) will be provided. The receive and transmit data bus is 2 bit wide, the signal TX_ER is not supported. CRS and RX_DV are collapsed to a single signal CRS_DV. The collision signal COL is completely removed, collisions will be detected as soon as data will be sent and received at the same time. In order to avoid collisions the variable CRS_AND_TX_EN_INFER_COL (see **Chapter 4.2.2.1**) has to be set.

The following figure illustrates the behavior of the CRS_DV signal in the EFM application (the pure CRS signal might still be asserted, even when no data will be sent to the MAC). Collapsing of CRS and RX_DV will be done in defining a CRS phase and RX_DV phase on di-bit boundaries within a nibble when transmitting data to the MAC. In this case, the CRS part will be driven on cycles of REF_CLK which present the first di-bit of a nibble. Consequently, the RX_DV part will be driven on cycles of REF_CLK which present the second di-bit of a nibble.

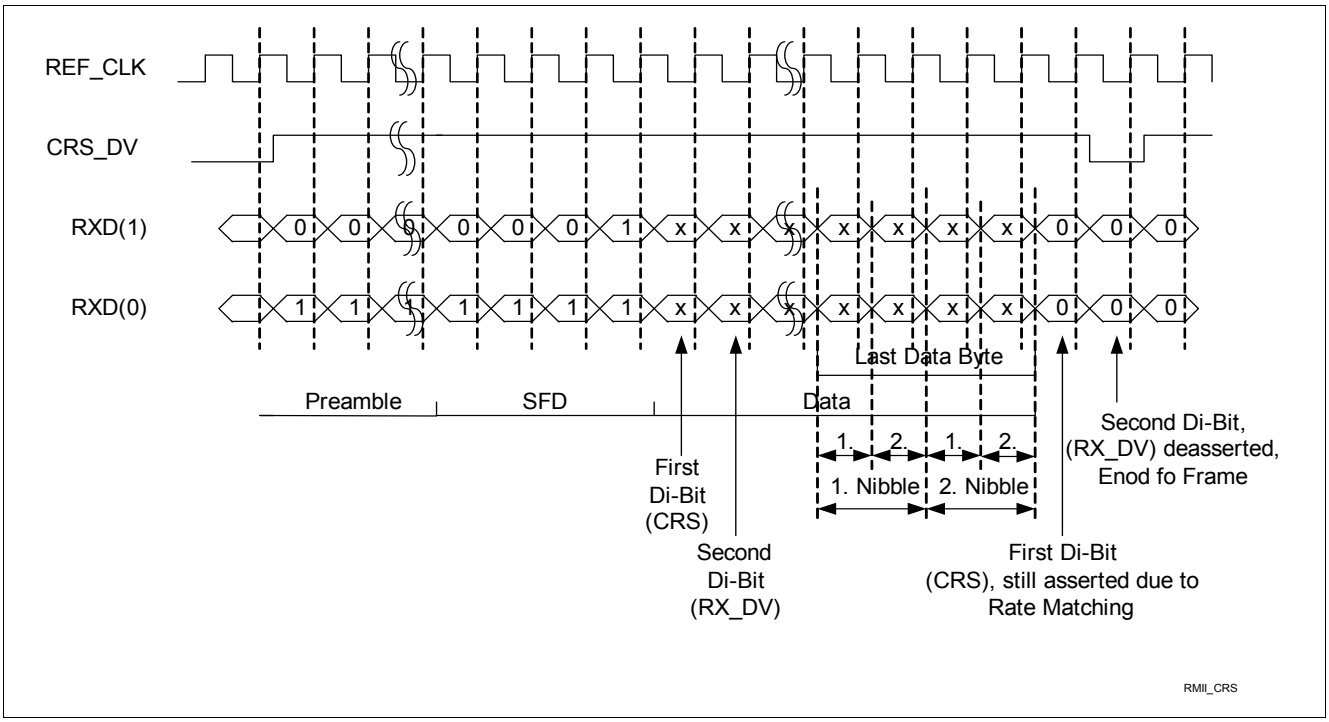


Figure 69 CRS Signal in RMII Mode

SS-SMII Mode

The SS-SMII interface is implemented according to the Serial-MII Specification, Revision 2.1. In SS-SMII mode the CLOCK (f = 125 MHz) has to be provided , RXD will be clocked with RX_CLK (driven by SOCRATES™-4e), TXD will be clocked with TX_CLK, which is driven by the MAC.

Data will be sent serially, where 8 bit of information will be sent in 10 bit segments. The beginning of this 10 bit segment is marked by the SYNC signal (2 SYNC signals - TX_SYNC, RX_SYNC - in SS-SMII mode). As in RMII mode, the collision signal COL is completely removed, collisions will be detected as soon as data will be sent and received at the same time (TX_EN and CRS active the same time). In order to avoid collisions the variable CRS_AND_TX_EN_INFER_COL (see **Chapter 4.2.2.1**) has to be set.

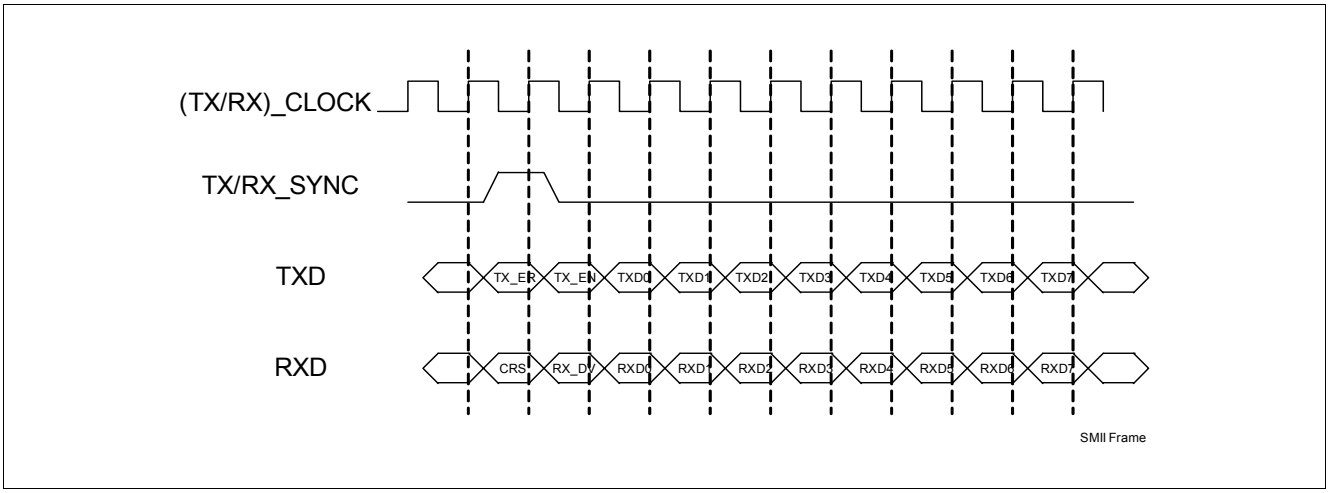


Figure 70 SS-SMII Frame Structure

Rate Matching will be done using the CRS signal, in this case segments with the CRS bit set and the RX_DV bit not set will be sent to the MAC. If RX_DV = ’0’ the segment carries the following information:

Table 35 RXD Encoding, RX_DV = 0

Bit	Meaning
RXD0	RX_ER from previous frame
RXD1	Speed = 1 (100 Mbit/s)
RXD2	Duplex = 0 (Half Duplex)
RXD3	Link = current link status (0 = Down, 1 = Up
RXD4	Jabber = 0 (no Jabber)
RXD5	Upper Nipple = 1 (valid)
RXD6	False Carrier = 0
RXD7	1

Minimum Received Preamble Length

In order to decode the Start of Frame delimiter on TXD correctly the different interface require the following minimum preamble length (the entire SFD character has to be present in all modi):

Table 36 Minimum Preamble Length at Different MII Interfaces

xMII Interface Type	Preamble Length [byte]	Note
MII	0	Only SFD sufficient
RMII	1	Min. 1 byte of preamble required
SS-SMII	1	Min. 1 byte of preamble required

MDIO Interface

The MDIO Interface consists out of 2 pins, the MDC (Management Data Clock, f_{MAX}(MDC)=2,5 MHz) and the MDIO (Management Data IN/OUT). The MDC will be driven by the MAC, the MDIO is an In/Out pin. This interface is available in all different MII flavours, but only once implemented for the entire chip. The MDIO interface will be used for configuration and management purposes of the PHY.

The MDIO frame structure is implemented according to Clause 45 (Extension to Clause 22) of IEEE 802.3-2004. 4 different type of frames are supported:

- Address Frame
- Write Frame
- Read Frame
- Post-Read-Increment-Address Frame

All these frames use an Start of Frame indicator (ST) of 01 (in contrast to Clause 22 management frames, which use a Start of Frame Indicator of 00).

For all accesses besides Post-Read-Increment-Address Accesses to the SOCRATES™-4e working in PHY mode 2 frames are necessary. First the address frame will be sent. This frame specifies the device address (PRTAD), the MMD to be accessed (DEVAD) and the register within the MMD (Address). This address will be stored at the internal address register. Write frames also contain the device address, the MMD and the contents to be written to the register address stored in the internal address register. Read frames contain the device address and the MMD, afterwards the MAC tristates the output and the PHY can provide the contents of the register addressed by the internal address register. When using Post-Read-Increment-Address frame the address frame can be omitted, consecutive registers will be read when sending these frames.

The device address consists out of 5 bit, the 2 LSB specify the channel to be accessed within SOCRATES™-4e.

Table 37 Mapping of the Device Address to SHDSL Channels

Channel Number	Device Address (1:0)
SHDSL Channel 0	00
SHDSL Channel 1	01
SHDSL Channel 2	10
SHDSL Channel 3	11

The 3 LSB of the device address can be assigned by pin strapping (pins PHYADDR2-4). So in total 8 SOCRATES™-4e can be connected to 1 MDIO interface.

The SOCRATES™-4e supports the MMD 1 (PMA/PMD), MMD 3 (PCS) and MMD 6 for realizing the EFM functionality. Additionally register 5 and 6 of all other MMD is supported as well indicating the supported MMD's as well as the vendor specific MMD's. Both vendor specific MMD's (MMD 30, MMD 31) are supported by the SOCRATES™-4e, this part contains is only used for debugging purposes. The following table lists some common register available in all MMD's and their content:

Table 38 Common Register of different MMD's

Register	Register Address	Content (15:0)[x]	Meaning
Device In Package	5	004A	PMA/PMD, PCS and TC present in package
Device in Package	6	00C0	Vendor Specific MM 1 and 2 present in package
Device ID	2	0302	Combination of OUI and device ID of SOCRATES™-4e
Device ID	3	6091	

5.2.1.2 xMII Interface in MAC Mode

Figure 68 shows the xMII interface in MAC mode, **Table 14** lists the available pins and their direction.

In MAC mode the different MII flavours MII and SS-SMII are available (refer to **Table 34** for the number of supported interfaces per flavor). The interface can be operated in half- and in full-duplex mode and need to be configured during start up. When operating in half-duplex mode the same interface signals as described in the previous chapter (see **Chapter 5.2.1.1**) are available. When connected and Ethernet PHY, no Rate Matching needs to be done, so all interface types behave as standardized.

Full Duplex Mode

When working in full duplex mode no collision can occur. Therefore the inputs COL and CRS in MII mode (or the derived signals in SS-SMII mode) are not monitored.

Minimum Preamble Length

In order to successfully detect the start of delimiter on RXD, a minimum length of preamble is required in different mode (see **Table 36**).

MDIO Interface

The SOCRATES™-4e also supports a MDIO interface in MAC mode. Refer to the MDIO Interface section of the previous chapter for a general description and functionality of this interface. In MAC mode the SOCRATES™-4e drives the MDC. The supported frame format is according to Clause 22 of IEEE 802.3-2004. For accessing a PHY 1 frame is sufficient. The start of frame indicator (ST) for these frames is 00 in order to differentiate from MDIO frames according to Clause 45. Access to the 32 register of the MII management register set of an Ethernet PHY is supported.

5.2.1.3 SS-SMII Interfaces in Cascading Mode

In case several SOCRATES™-4e should share the same SYNC and CLK signals, the SOCRATES™-4e offers an option for synchronizing the different SYNC and CLK signals as shown in **Figure 71**. Cascading is supported in MAC and PHY mode. In MAC mode, Tx and RX signals are just “renamed” compared to the figure in PHY mode. The timing is critical (125 MHz signals), and should be balanced as indicated. Optional capacitors could be used for fine tuning of timing between SYNC_i and CLOCK_o to prevent setup/hold time violations. However, SOCRATES™-4e also offers an option to modify this timing internally via register settings utilizing internal delay elements. Please be aware, that temperature and voltage variations might have an impact on the timing.

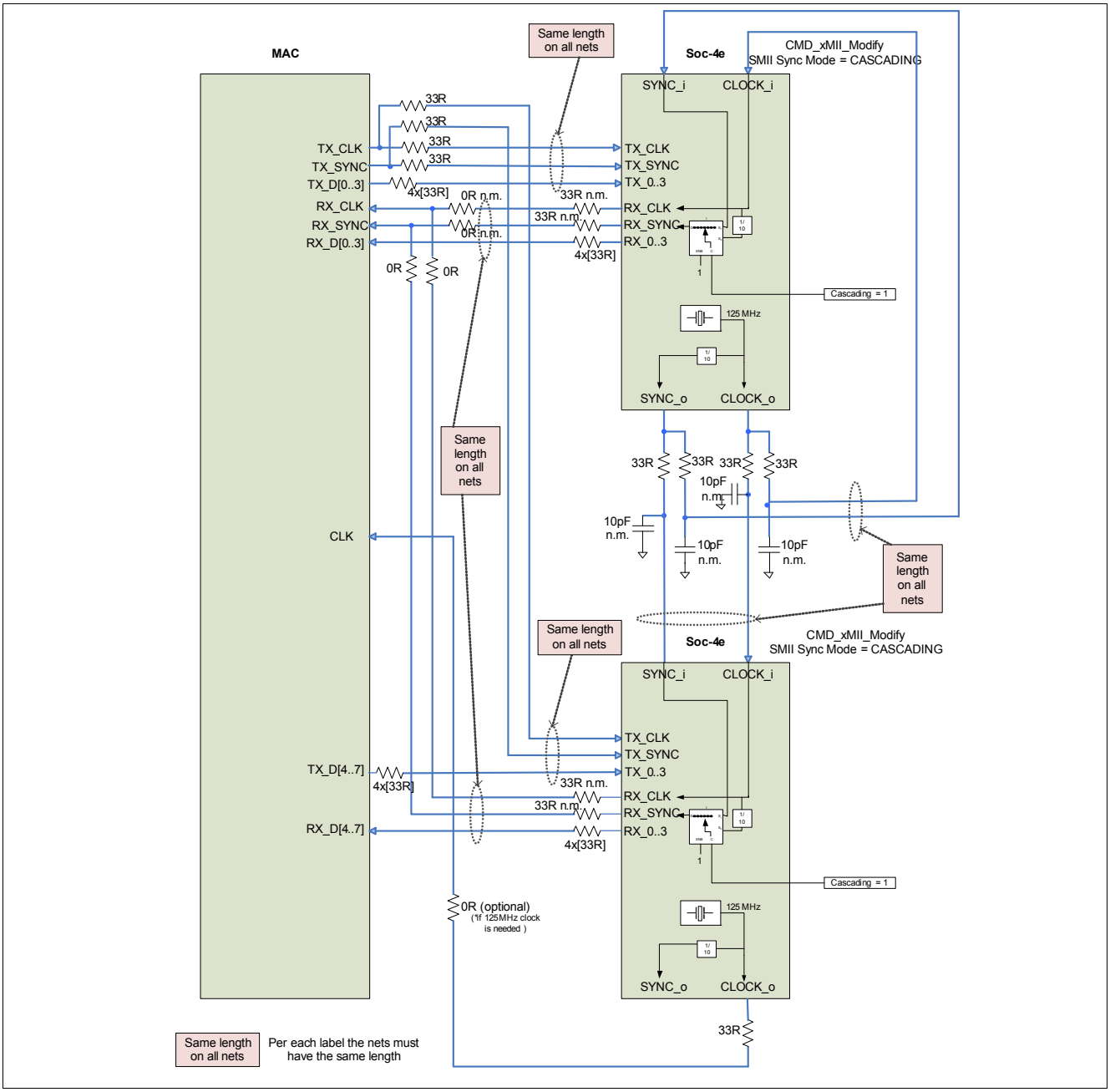


Figure 71 SS-SMII Cascading Mode in PHY Mode

5.2.2 UTOPIA/POS-PHY Interface

The other mode of System Interface 1 is the UTOPIA/POS-PHY mode. **Table 9** lists the available pins and their direction.

5.2.2.1 Utopia Mode

The UTOPIA interface enables the transfer of ATM cells via ATM based backplanes. This interface type has been implemented according to the standard AF-PHY-0039.000 from the ATM Forum for UTOPIA Level 2, Version 1.0. The interface receiver is located in the device transmit direction and the interface transmitter in the device receive direction, only SLAVE mode is supported.

Optional supported features are:

- Parity generation and evaluation
- Support of Multiplexed Status Polling operation

A bus width of 8 and 16 bit is supported and need to be configured using the pins TCCONF(2:0) The pins URXDATA[15:8] and UTXDATA[15:8] are not available in 8-bit data bus width operation and used for different functions.

Data Formats

The standard cell formats with 53 octets in the interface 8-bit mode and with 54 octets in the interface 16-bit mode are shown in **Figure 72**. The content of the fields UDF1 and UDF2 is not defined at the interface.

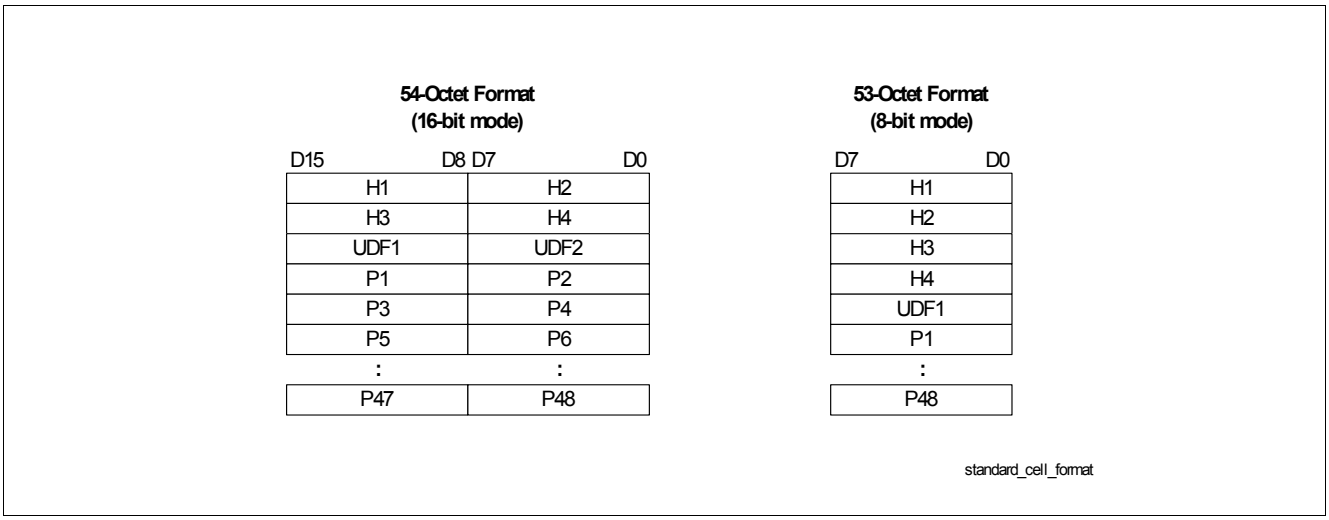


Figure 72 Standard Cell Format

Interface Address Handling

The system interface 1 itself can be disabled and enabled. If it is disabled all interface pins are tristated. Additionally every port can be enabled/disabled.

In total 5 ports are supports (1 per line, 1 port for device configuration). Every port has a default UTOPIA address assigned, which corresponds to the line number. The port address for the device configuration port is 4_H, port addresses are identical for transmit and receive direction. The default address can be overwritten.

Disabled ports respond to a poll request with tristate.

Backpressure Generation

For each enabled system interface port in device transmit direction, a buffer for at least 2 ATM cells is provided. The port specific buffer prevents head-of-line-blocking, that means blocked interface ports can not block data transfer on other ports. When the port specific buffer reaches its limit backpressure is generated by deasserting the TXCLAV signal when the related address is polled.

Handling of UDF Field

The fields UDF1 and UDF2 of ATM cells are not defined at the interface. The transmitter located in device receive direction generates these fields but the value is not standardized, but can be configured. The receiver located in device transmit direction doesn't evaluate UDF1 and forwards it without any changes to other internal protocol units. UDF2 is discarded without any evaluation by the receiver.

Parity Generation and Evaluation

The parity handling is defined as optional in 16-bit mode operation according to AF-PHY-0039.000.

The interface transmitter generates an odd parity signal over the 16 data signals at pin RXPRTY. This pin is tristated when the respective data signal pins are tristated. The parity generation is active if the interface itself is enabled.

The interface receiver generates an odd parity over the 16 data signals in 16-bit mode, over 8 data signals in 8-bit mode and compares the generated value with the received value from the interface. In case of mismatch the related ATM cell can be discarded and an interrupt can be generated. The interrupt generation in case of mismatches is on/off-switchable by the user. The cell discarding in case of mismatches is on/off-switchable by the user and switched off per default.

Error Handling

The Start of Cell (TXSOC) signal is checked. Received Cell with a missing SoC signal will be internally dropped.

Loop Backs

2 different loops will be offered per port. Both loops are always enabled/disabled together.

Ingress Loop Backs

An interface ingress loop back can be enabled/disabled per port. The ingress loop back mirrors the data stream from the receiver (pin UTXDATA) tot the transmitter (pin URXDATA). The loop back is non-transparent.

Egress Loop Back

An interface egress loop back can be enabled /disabled per port. The egress loop back mirrors the to-be-transmitted data stream from the transmitter to the receiver. This loop is non-transparent.

5.2.2.2 POS-PHY Mode

The POS-PHY interface enables the transfer of packets via packet based backplanes. It operates in SLAVE Mode. The interface type Level 2 has been implemented according to PMC-971147 Issue 5 of the Saturn group. The interface receiver is located in the device transmit direction and the interface transmitter in the device receive direction.

Optionally supported is parity generation and evaluation.

Data Format

The packet format is shown in the following figure. The format of expected packet are plain Ethernet frames without any encapsulation. Furthermore it is expected that no preamble/SFD is prepended in device transmit direction. No preamble/SFD will be sent in device receive direction over POS-PHY interface. ATM cells can be transported via POS-PHY as well. In this case 54 octets with 2 UDF fields as described in **Figure 72** are expected.

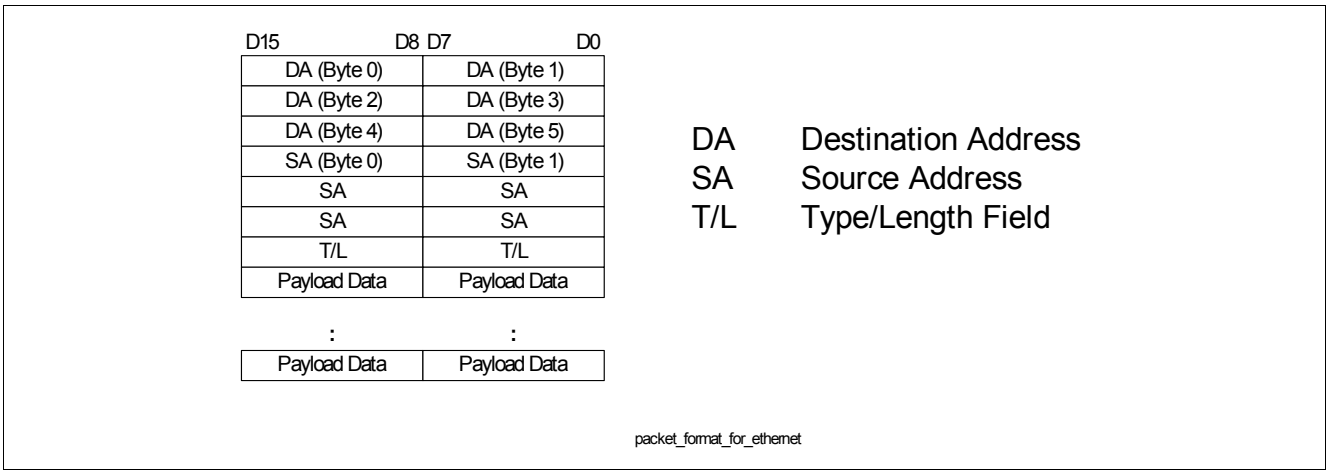


Figure 73 Frame Format Ethernet Frames

Interface Address Handling

The system interface 1 itself can be disabled and enabled. If it is disabled all interface pins are tristated. Additionally every port can be enabled/disabled.

In total 5 ports are supports (1 per line, 1 port for device configuration). Every port has a default POS-PHY address assigned, which corresponds to the line number. The port address for the device configuration port is 4_H, port addresses are identical for transmit and receive direction. The default address can be overwritten.

Disabled ports respond to a poll request with tristate.

Backpressure Generation

For each enabled system interface port (identified by interface address) in device transmit direction, a buffer for at least 2 times 64 octets is provided. The port specific buffer prevents head-of-line-blocking, that means blocked interface ports can not block data transfer on other ports. When the port specific buffer reaches its limit backpressure is generated by deasserting the packet available signal when the related address is polled.

Threshold Based Handling

The interface receiver forwards the data of the port specific FIFO to an internal buffer, but it doesn't initiate data transmission before an end of packet flag has been stored in the port specific FIFO for avoiding FIFO underflow. This applies to all ports

The interface receiver has a programmable threshold defined in terms of the number of bytes available in the port specific FIFO for the deassertion of the Packet Available flags (STPA signal). The user-programmable Almost Full Level or *Packet Available Deassertion Fill Level* is common for all interface ports and can be set in the following range. This parameter should be adjusted to characteristics of the connected device (how fast does the device react on deassertion of STPA).

- Full level minus n bytes with n = 2, 6, 10, ...50, 54 or 58

For the interface transmitter, the burst transfer size can be tuned by providing a programmable Almost Full Level or Packet Available Assertion Fill Level. The user-programmable value is common for all interface ports and can be set in the following range.

- Empty Level plus n bytes with n = 32, 36, ... 56, 60, 64 (this threshold should be set to 64)

Parity Generation and Evaluation

The parity handling is defined as optional in 16-bit L2 mode operation according to PMC-971147 Issue 5. If supported odd parity as well as even parity must be supported.

The interface transmitter generates a user-selectable odd or even parity signal over the 16 data signals at pin RXPRTY. This pin is tristated when the respective data signal pins are tristated. The parity generation is active if the interface itself is enabled.

The interface receiver generates a user-selectable odd or even parity over the 16 data signals and compares the generated value with the received value from the interface. In case of mismatch the handling of the related packet depends has the following options:

- On/off-switchable interrupt generation in case of mismatches
- On/off-switchable error marking of packets with mismatch

In L2 mode, the user can determine the handling for mismatches in following way.

Error Handling

If the interface transmitter gets an error-marked packet from the line side it will be also sent out error-marked on the interface.

If the interface receiver gets an error-marked packet from the interface it will be also sent out error-marked to internal units.

If the receiver detects interface protocol violations the related packet is forwarded error-marked. Such an interface protocol violation is for example when the receiver gets a second Start of Packet SOP indication before getting an End of Packet EOP indication. In this case, the packet data related to the first SOP is forwarded error-marked to the line side and packet data related to the second SOP is treated non-errored as long as no other error occurs.

Loop Backs

See respective Loop back section in UTOPIA interface description (see **Chapter 5.2.2.1**)

5.3 System Interface 2 - TDM Interface

The TDM interface can be configured to different modes which are described for one SHDSL link in the following sub clauses.

5.3.1 T1-, E1-, IOM-2 and Bit Serial Framing

For T1 framing the frame sync pulses (TFSC/RSFC) are directly related to the F-bit according to **Figure 74**. One T1 frame consists of 192 data bits plus F-bit.

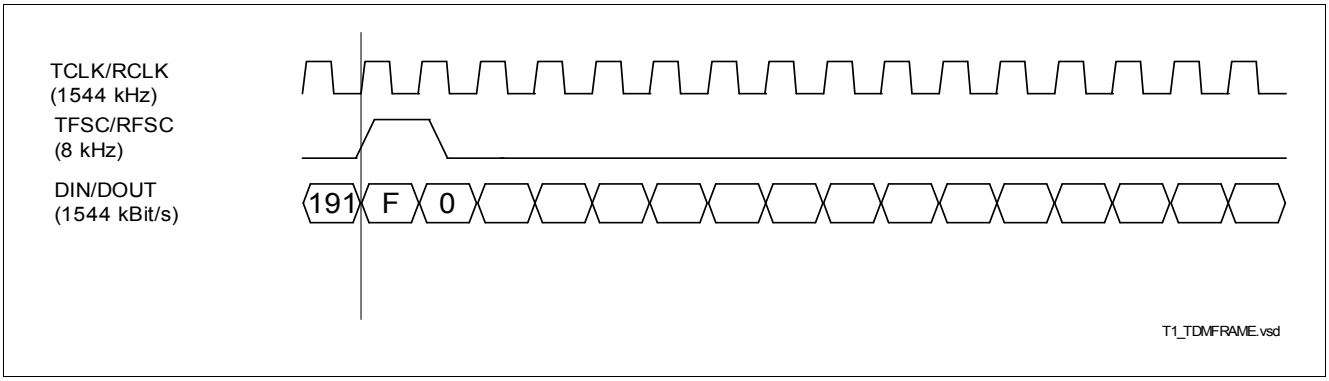


Figure 74 T1 Framing

For E1 framing the frame sync pulses (TFSC/RSFC) are directly related to the 1st data bit according to **Figure 75**. One E1 frame consists of 256 data bits.

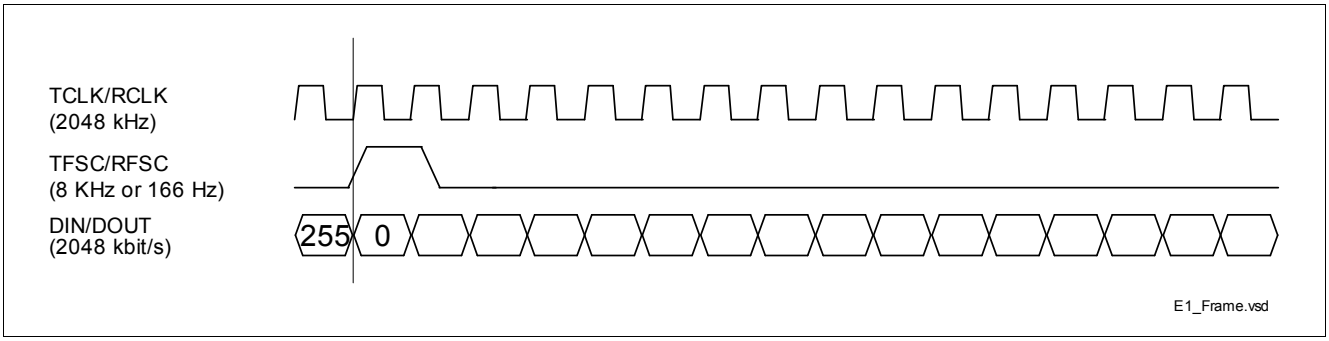


Figure 75 E1 Framing

For IOM-2 framing the frame sync pulses (TFSC/RSFC) are directly related to the 1st data bit according to **Figure 76** and **Figure 77**. One IOM-2 frame consists of n x 32 data bits where n stands for the number of ISDN connections to be realized by the SHDSL link. According to ITU-T G.991.2 (2004) the maximum number for n is 6, i.e. up to 6 ISDN connections can be realized by one SHDSL link.

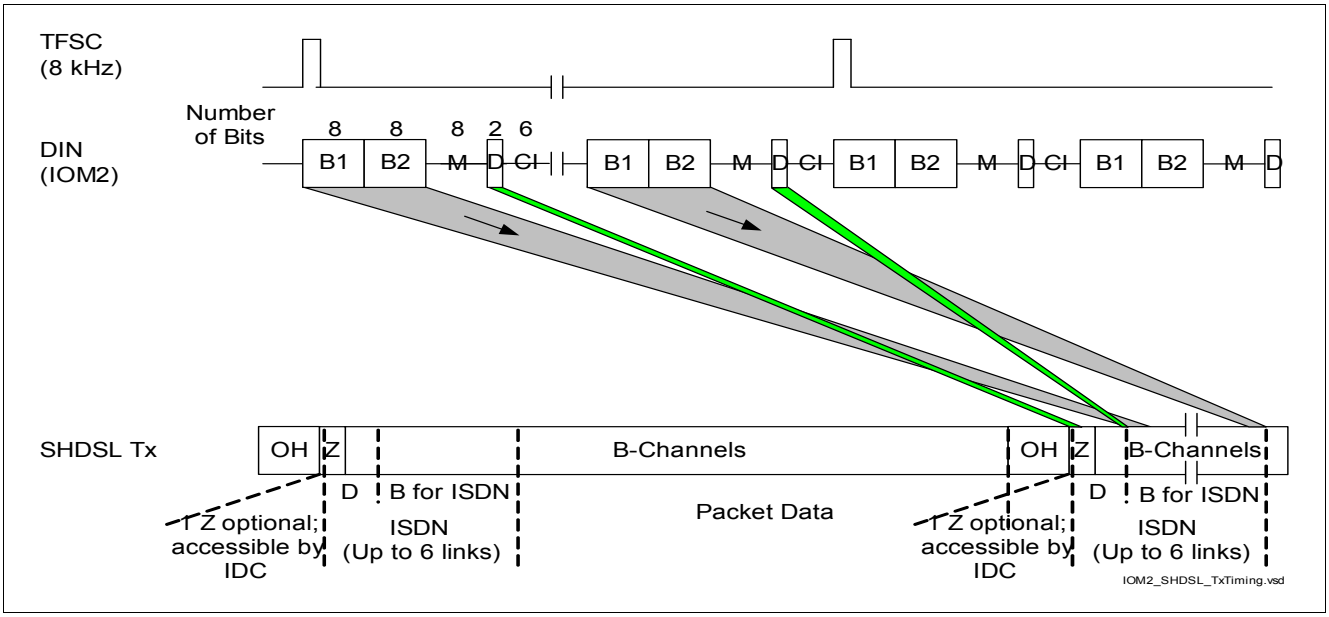


Figure 76 IOM-2 Framing, Tx Direction

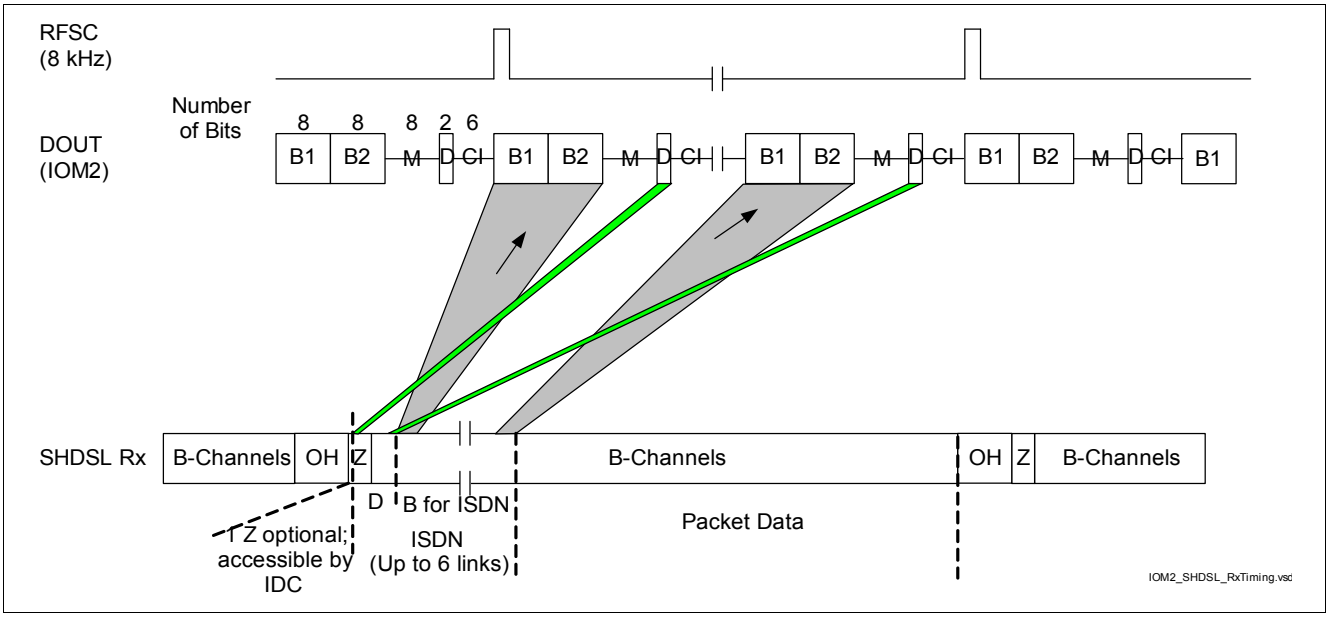


Figure 77 IOM-2 Framing, Rx Direction

For Bit Serial framing no frame sync pulses (TFSC/RSFC) are required (see **Figure 78**). One Bit Serial frame consists of the number of payload bits transferred via one sub-block of the SHDSL frame.

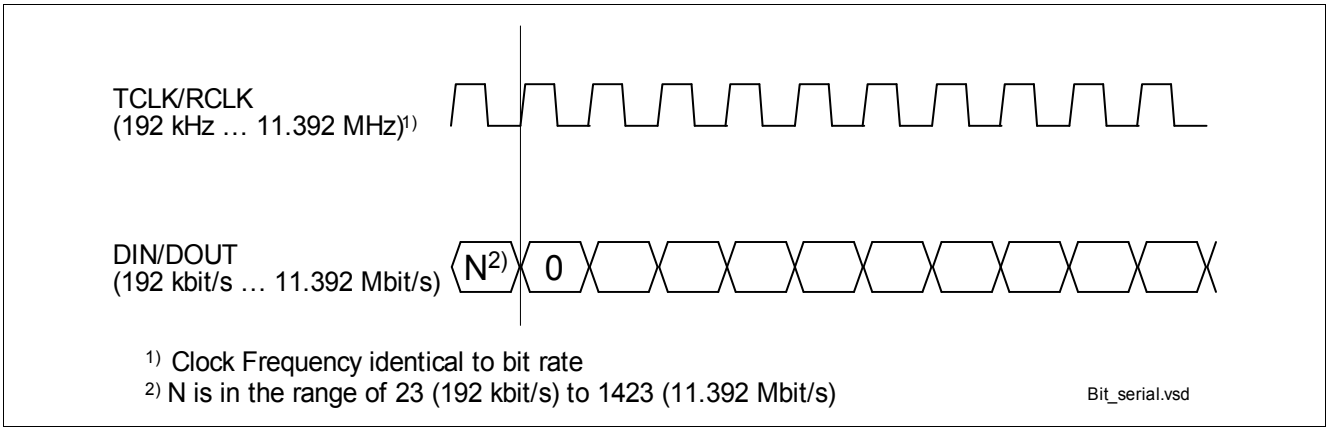


Figure 78 Bit Serial Framing

5.3.2 Framing with Z-Bits and Offset

For applications with TDM frames containing Z bits these Z bits can either be transferred via the TDM interface in not octet aligned as shown in **Figure 79** mode or octet aligned mode as shown in **Figure 80**.

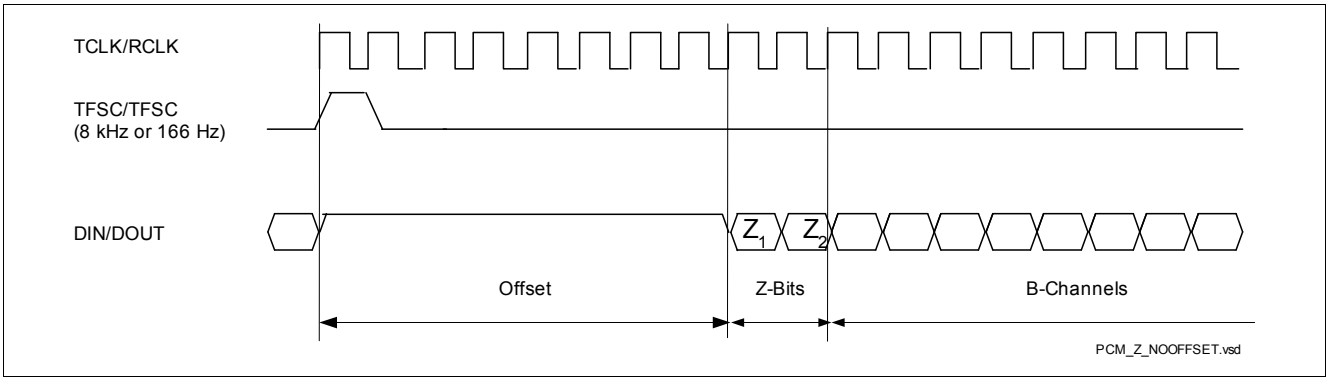


Figure 79 Example for Byte Offset, Z Bits Not Octet Aligned

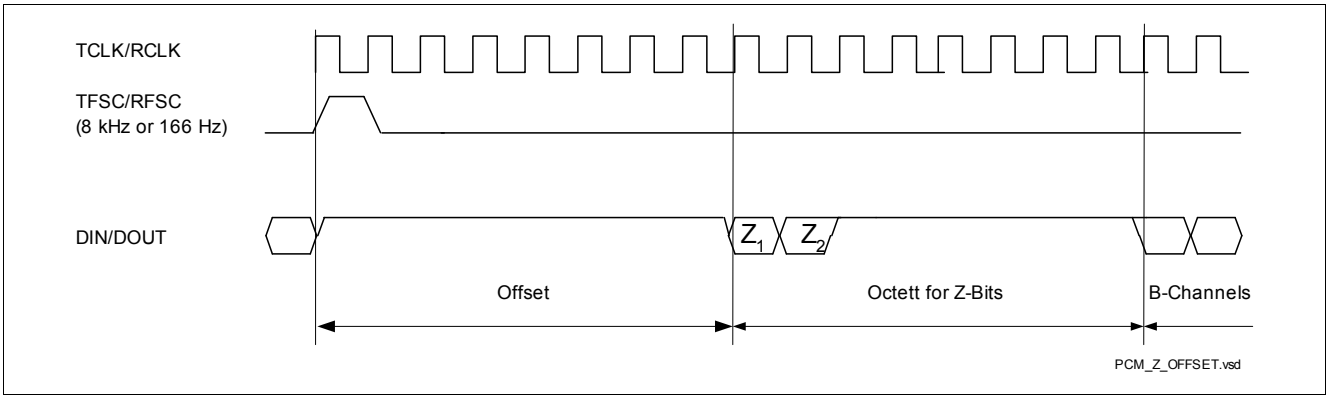


Figure 80 Example for Byte Offset, Z Bits Octet Aligned

5.3.3 TDM in Bus Mode

If more than one SHDSL link is active with TDM data transferred to each of those links the TDM interface can be configured to the TDM bus mode.

In bus mode more than one SOCRATES™-4e can be connected to one TDM interface with the restrictions that the number of connected SHDSL links must not exceed 16 and that the bit rate must not exceed 16.384 Mbit/s.

The data is transferred bitstream interleaved via the TDM interface, i.e. payload data for one SHDSL link is a consecutive bit stream on the TDM interface (see.**Figure 81**)

The payload data for one SOCRATES™-4e is transferred via TDM consecutively, i.e. 1st part of this consecutive TDM data is mapped to SHDSL link 0, last TDM channel is mapped to link 3.

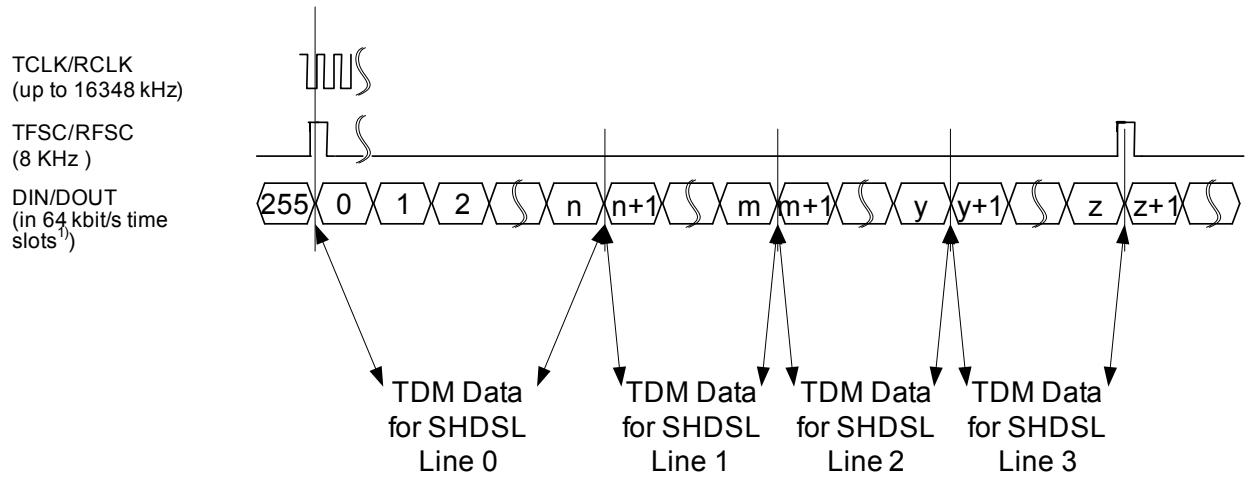


Figure 81 TDM Bus Mode

5.4 JTAG Interface of the SOCRATES™-4e

The boundary scan functionality is implemented according to IEEE 1149.1 boundary scan standard. With boundary scan the SOCRATES™-4e can be accessed for testing purposes.

The JTAG interface of the SOCRATES™-4e uses a 5-pin test access port (TAP) as specified in **Chapter 2.2.7**. The identification register can be serially read out via pin TDO (Test Data Output). It contains the following value:

Table 39 Boundary Scan ID

Version	Device Code	Manufacture Code		Output
0001	0000 0000 1010 1101	0000 1000 001	1	-> TDO

6 Operational Description

The SOCRATES™-4e is configured and maintained by the Host via the microprocessor interface. The communication between Host and SOCRATES™-4e is based on messages.

7 Electrical Characteristics

This chapter describes the DC and AC characteristics of the SOCRATES™-4e.

7.1 Absolute Maximum Ratings

Table 40 Absolute Maximum Ratings

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Max. Junction Temperature	T_J	-55	–	125	°C	–
Core supply voltage	V_{DD}	-0.5	–	1.7	V	–
I/O supply voltage	V_{DDP}/V_{DDA}	-0.5	–	4.0	V	V
Voltage on any pin	V_{max}	-0.5	–	4.0	V	–
Maximum DC current on any pin		-10	–	10	mA	–
ESD robustness HBM: 1.5 kΩ, 100 pF	$V_{ESD,HBM}$	–	–	2000	V	According to EIA/JESD22-A114-B
ESD robustness	$V_{ESD,SDM}$	–	–	500	V	According to ESD Association Standard DS5.3.1 - 1999

*Note: The maximum voltage difference must not exceed 4.0 V in any case (i.e. Supply voltage = 4.0 V and Input Voltage = -0.5 V is **not** allowed!)*

Attention: Absolute Maximum Ratings are stress ratings only, and functional operation and reliability under conditions beyond those defined in the normal operating conditions is not guaranteed. Stresses above the maximum ratings are likely to cause permanent damage to the chip

7.2 Operating Range

Table 41 Operating Range

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Ambient temperature under bias PEF	T_A	-40	–	85	°C	–
Core supply voltage	V_{DDD}	1.42	1.5	1.58	V	–
I/O supply voltage	V_{DDP} V_{DDA}	3.13 3.13	3.3	3.47 3.47	V V	–
Ground	V_{SS}	0	0	0	V	–

Note: In the operating range, the functions given in the circuit description are fulfilled.

7.2.1 Recommended Operation Conditions

The SOCRATES™-4e meets all standards for operation in indoor and outdoor environments in communication systems.

7.3 Power Up Sequence

To avoid damage of the SOCRATES™-4e during power up one of the following rules should be applied:

Use the following sequence for biasing:

1. Core Voltage
2. Pad Voltage not before Core Voltage
3. Signal Voltage not before Pad Voltage

If this sequence does not meet your requirements make sure that

- The inverse current per signal pad < 10 mA
- The current per supply domain < 100 mA

7.4 Line Overload Protection

The maximum input current for the loop interface (under over-voltage conditions) for the loop interface is given as a function of the width of a rectangular input current pulse. For the destruction current limits refer to **Figure 82**.

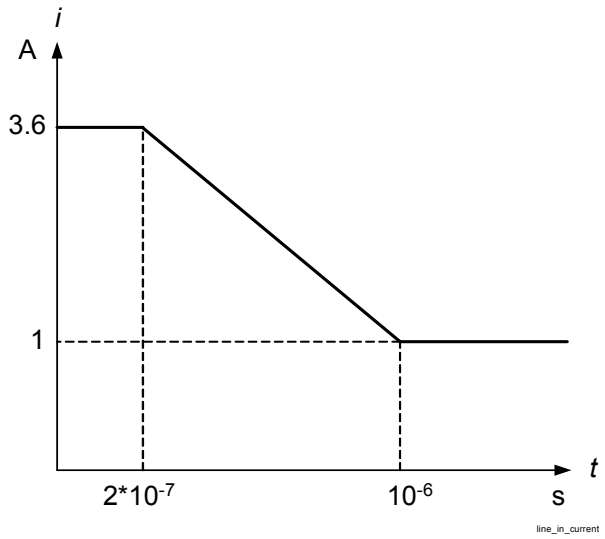


Figure 82 Maximum Line Input Current

7.5 DC Characteristics

Table 42 DC Characteristics

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Input low voltage	V_{IL}	-0.4	–	0.8	V	–
Input high voltage	V_{IH}	2.0	–	3.6	V	–
Output low voltage	V_{OL}	–	–	0.4	V	Buffer A: 20 mA
		–	–	0.4	V	Buffer B: 10 mA
		–	–	0.4	V	Buffer C: 5 mA

Table 42 DC Characteristics (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Output high voltage	V_{OH}	2.4	–	–	V	Buffer A: -20 mA
		2.4	–	–	V	Buffer B: -10 mA
		2.4	–	–	V	Buffer C: -5 mA
Avg. power supply current V_{DDD} supply: 192 kbit/s 1536 kbit/s 2304 kbit/s . V_{DDP} supply: V_{DDA} supply:	I_{CC} (AV)	–	 tbd tbd tbd tbd 400	–	mA	V_{DDD} = 1.5 V, $V_{DDP/DDA}$ = 3.3 V, T_A = 25 °C All 4 channels active
Input leakage current	I_{IL1}	–	–	10	µA	Test condition V_{DDP} = 3.3 V, V_{SS} = 0 V, all other pins are floating V_{in} = 0 V
	I_{IL2}	–10	–	-	µA	Test condition V_{DDP} = 3.3 V, V_{SS} = 0 V, all other pins are floating V_{in} = 3.3 V
Output leakage current	I_{OL1}	–	–	10	µA	Test condition V_{DDP} = 3.3 V, V_{SS} = 0 V, V_{out} = 0 V
	I_{OL2}	–10	–	-	µA	Test condition V_{DDP} = 3.3 V, V_{SS} = 0 V, V_{out} = 3.3 V
Pull Up Current[1)]	I_{PUB}	11	–	34	µA	Test condition 0 V

1) Not subject to production test - verified by design/characterization.

Note: The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread.

7.6 Capacitances

T_A = 25 °C, 3.3 V ± 5% and 1.5 V ± 5%, f_c = 1 MHz, unmeasured pins grounded.

Table 43 Capacitances

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Input Capacitance I/O Capacitance	C_{IN} $C_{I/O}$	–	–	5	pF	All pins except XTAL1 and analog front ends
Output Capacitance against V_{SS}	C_{OUT}	–	–	5	pF	All pins except XTAL2 and analog front ends
Input Capacitance	C_{IN}	–	–	7	pF	XTAL1
Output Capacitance against V_{SS}	C_{OUT}	–	–	7	pF	XTAL2

7.7 Specification of the Crystal

A crystal (parallel resonance) has to be connected to XTAL1 and XTAL2 which shall meet the following specification:

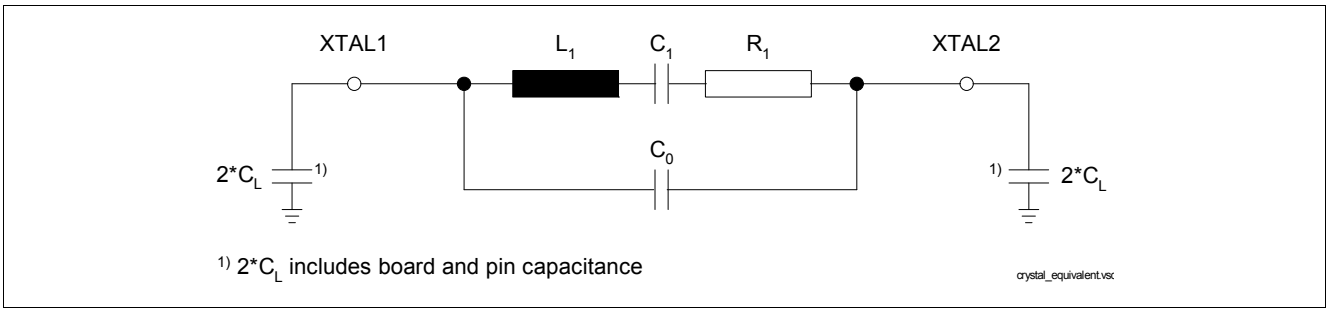


Figure 83 Equivalent Circuit for Crystal Specification

Table 44 Specification of the Crystal

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Nominal Frequency		–	20.48/ 40.96[1)]	–	MHz	–
Temperature Range		-40	–	+85	°C	–
Total Frequency Stability[2)]		-60	–	+60[3)]	ppm	–
Load Capacitance	C_L	–	tbd	–	pF	–
Shunt Capacitance	C_0	–	7	–	pF	–
ESR	R_1	–	–	tdb	Ω	–
Drive Level		0.1	–	–	mW	–

1) See **Chapter 4.3.2** for selection
2) This stability refers to the sum of all tolerances (general, temperature, aging)
3) ± 32 ppm are required for Internal Timed mode (see Chapter 5.4), ± 60 ppm are required for Loop Timed and External Timed mode (see Chapter 5.4) in order to meet the standards (ITU-T G.991.2, ITU-T G.991.2 (2004) and ETSI TS 101 524)

The crystal specifications shall meet the requirements given in **Table 44**.

7.8 Specification of the Transformer

For the specification of the transformer as well as recommended transformer types refer to the respective application note.

7.9 Reset Behavior

To reset the SOCRATES™-4e properly, a reset pulse with a minimum pulse width as defined in **Table 45** has to be applied. Spikes on the reset line of up to 10 ns are ignored.

The 20.48/40.96 MHz clock has to be provided for pin XTAL1 of the SOCRATES™-4e during reset.

The device address is latched in according to **Figure 84**.

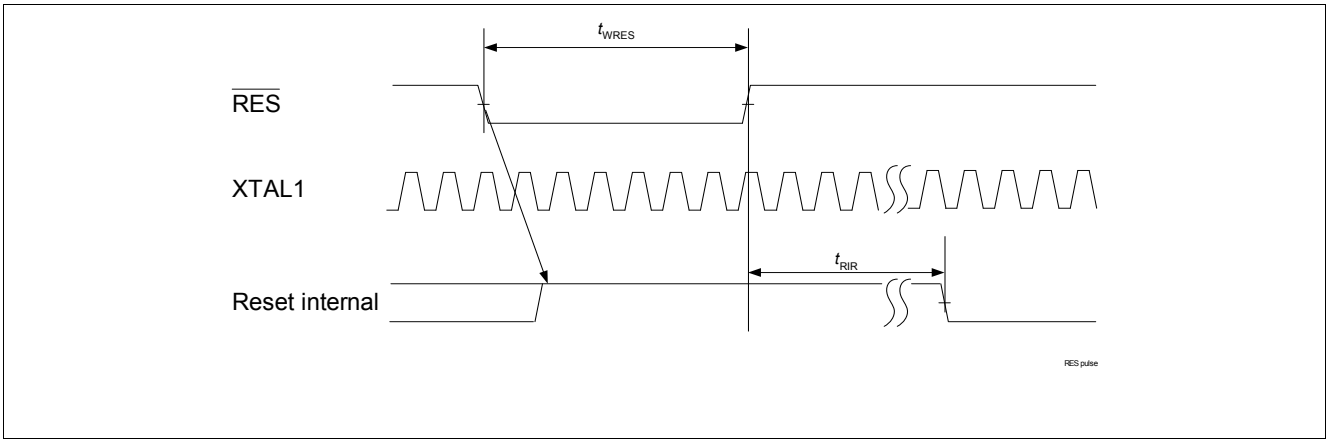


Figure 84 Reset Behavior

Table 45 Reset Behavior

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Pulse Width	t_{WRES}	150	–	–	ns	–
Internal Reset Release	t_{RIR}	–	–	200	µs	–
Device Address setup time	t_{AUXs}	50	–	–	ns	–
Device Address hold time	t_{AUXh}	200	–	–	µs	–

7.10 AC Characteristics

T_A = -40 to 85 °C, V_{DDA} / V_{DDP} = 3.3 V ± 5%, V_{DD} = 1.5 V ± 5%, V_{SS} = 0 V.

Inputs are driven to V_{IH} = 2.4 V for a logical 1 and to V_{IL} = 0.4 V for a logical 0. Timing measurements are made at 2.0 V for a logical 1 and at 0.8 V for a logical 0.

The AC testing input/output waveforms are shown in **Figure 85**. The load is according to **Figure 85** if not mentioned explicitly.

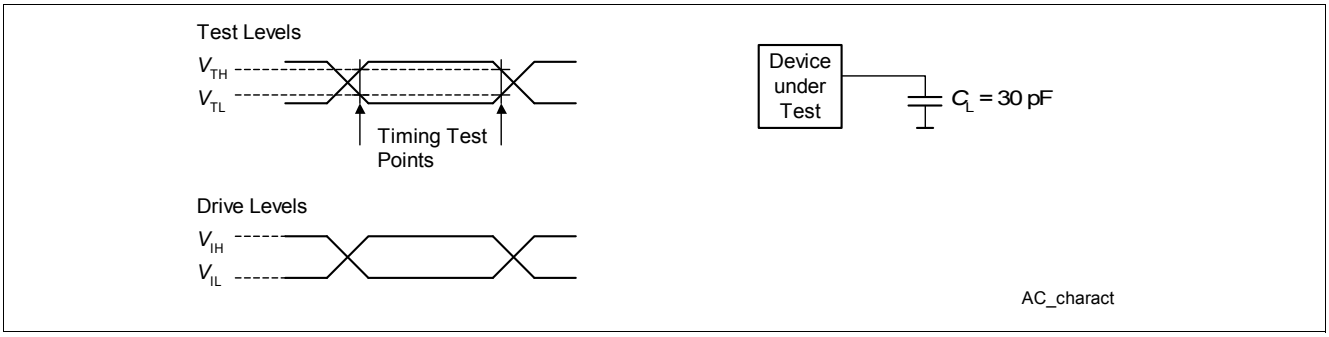


Figure 85 Input/Output Waveform for AC Tests

7.11 Timing Characteristics

This chapter describes the timing of the supported interfaces in detail.

7.11.1 Microprocessor Interface Timing

Due to the preliminary nature of this document all the timings given in this section are subject of change.

7.11.1.1 Intel Demux Interface Timing

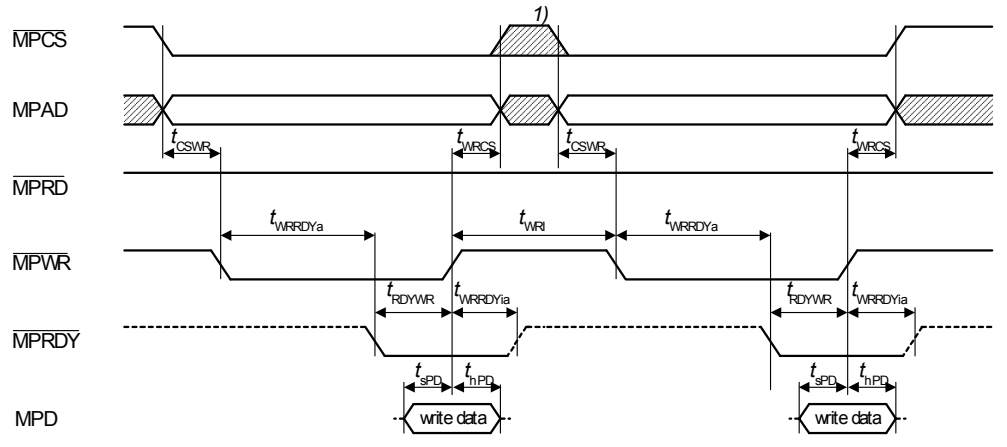


Figure 86 Intel Demux Write Timing

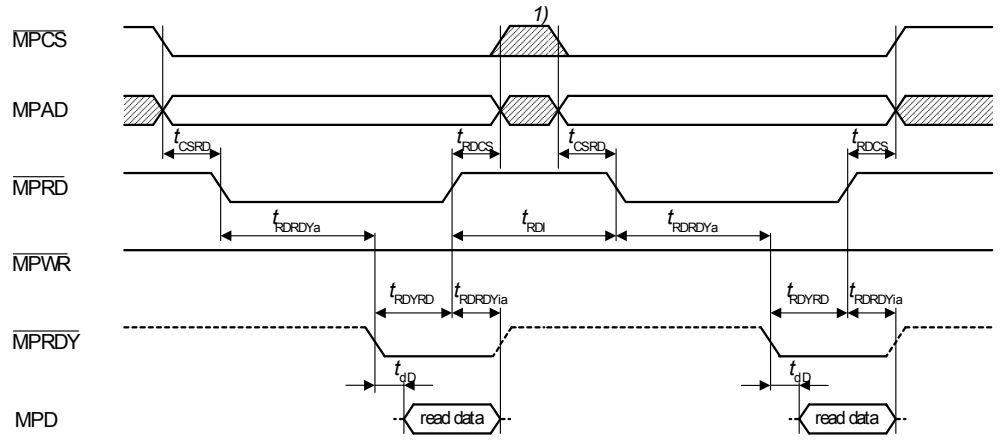


Figure 87 Intel Demux Read Timing

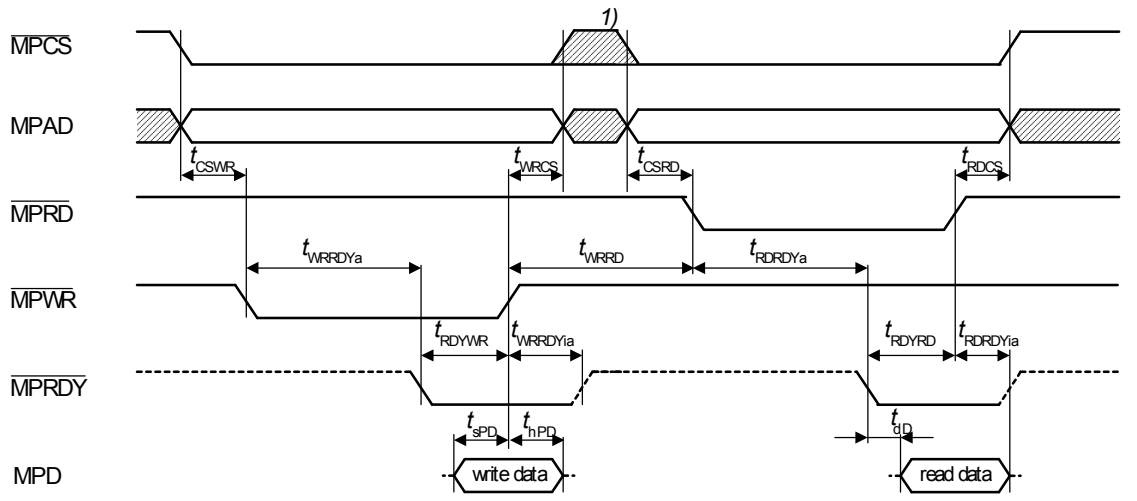


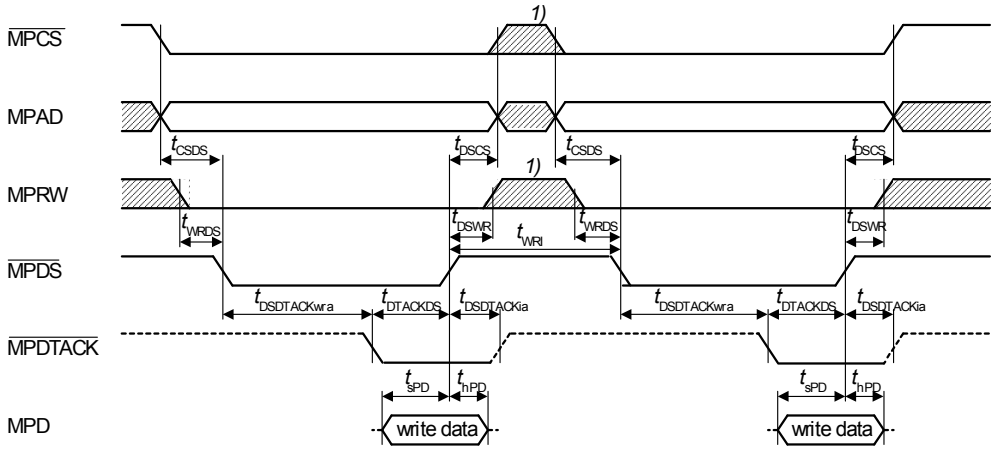
Figure 88 Intel Demux Write/Read Timing

Table 46 Intel Demux Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Write Interval	t_{WRI}	13	–	–	ns	–
$\overline{MPCS}$, MPAD setup to $\overline{MPWR}$	t_{CSWR}	0	–	–	ns	–
$\overline{MPRDY}$ active delay from $\overline{MPWR}$ active	t_{WRRDYa}	12	–	30	ns	–
$\overline{MPRDY}$ inactive delay from $\overline{MPWR}$ inactive[1)]	$t_{WRRDYia}$	5	–	15	ns	–
$\overline{MPWR}$ inactive delay from $\overline{MPRDY}$ active	t_{RDYWR}	0	–	–	ns	–
$\overline{MPCS}$, MPAD hold from from $\overline{MPWR}$ inactive	t_{WRCS}	0	–	–	ns	–
MPD setup time to $\overline{MPWR}$	t_{sPD}	10	–	–	ns	–
MPD hold time from $\overline{MPWR}$	t_{hPD}	10	–	–	ns	–
Read Interval	t_{RDI}	13	–	–	ns	–
$\overline{MPRD}$ to $\overline{MPWR}$ and $\overline{MPWR}$ to $\overline{MPRD}$ inactive time	t_{WRRD}	13	–	–	ns	–
$\overline{MPCS}$ setup to $\overline{MPRD}$	t_{CSRD}	0	–	–	ns	–
$\overline{MPRDY}$ active delay from $\overline{MPRD}$ active	t_{RDRDYa}	28	–	46	ns	–
$\overline{MPRDY}$ inactive delay from $\overline{MPRD}$ inactive[1)]	$t_{RDRDYia}$	5	–	15	ns	–
$\overline{MPRD}$ inactive delay from $\overline{MPRDY}$ active	t_{RDYRD}	0	–	–	ns	–
$\overline{MPCS}$, MPAD hold from $\overline{MPRD}$ inactive	t_{RDCS}	0	–	–	ns	–
MPD delay time from $\overline{MPRDY}$	t_{dD}	-8	–	0	ns	–

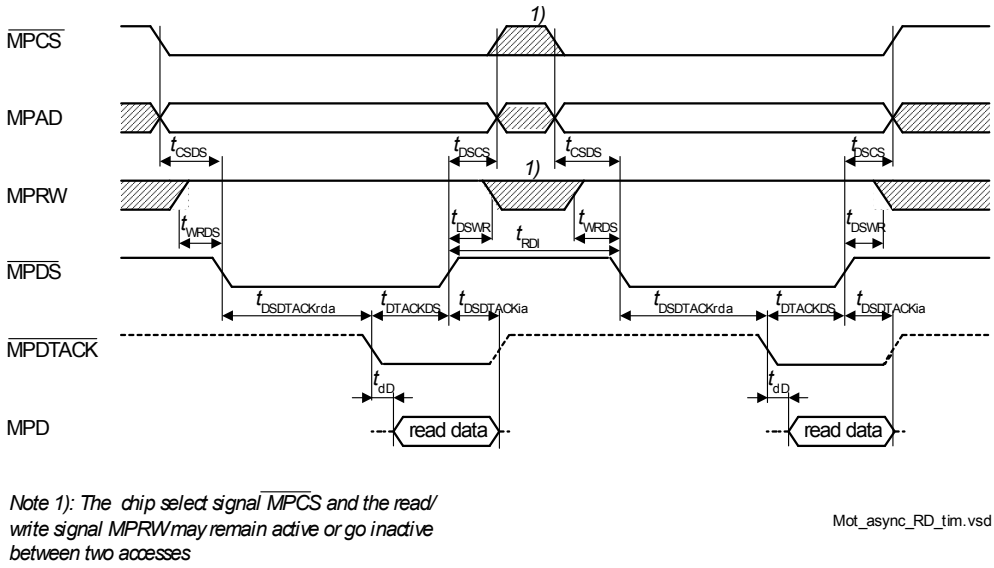
1) To hold this timing requirement pin $\overline{MPRDY}$ is driven ONE for 8 ns

7.11.1.2 Asynchronous Motorola Interface Timing



Note 1): The chip select signal $\overline{MPCS}$ and the read/write signal MPRW may remain active or go inactive between two accesses

Figure 89 Asynchronous Motorola Write Timing



Note 1): The chip select signal $\overline{MPCS}$ and the read/write signal MPRW may remain active or go inactive between two accesses

Figure 90 Asynchronous Motorola Read Timing

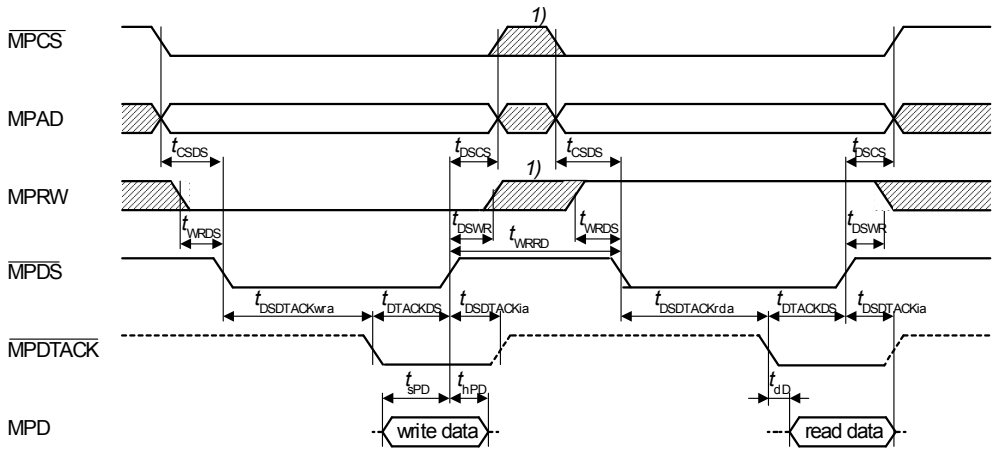


Figure 91 Asynchronous Motorola Write/Read Timing

Table 47 Asynchronous Motorola Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Write Interval	t_{WRI}	13	–	–	ns	–
$\overline{MPCS}$, $\overline{MPAD}$ setup to $\overline{MPDS}$	t_{CSDS}	0	–	–	ns	–
$\overline{MPWR}$ setup to $\overline{MPDS}$	t_{WRDS}	10	–	–	ns	–
$\overline{MPDTACK}$ active delay from $\overline{MPDS}$ active for write access	$t_{DSDTACKwra}$	12	–	30	ns	–
$\overline{MPDTACK}$ inactive delay from $\overline{MPDS}$ inactive	$t_{DSDTACKia}$	5	–	15	ns	–
$\overline{MPDTACK}$ active delay from $\overline{MPDS}$ active for read access	$t_{DSDTACKra}$	28	–	46	ns	–
$\overline{MPWR}$ hold from $\overline{MPDS}$ inactive	t_{DSWR}	10	–	–	ns	–
$\overline{MPDS}$ inactive delay from $\overline{MPDTACK}$ active	$t_{DTACKDS}$	0	–	–	ns	–
$\overline{MPCS}$, $\overline{MPAD}$ hold from $\overline{MPDS}$ inactive	t_{DSCS}	0	–	–	ns	–
MPD setup time to $\overline{MPDS}$	t_{sPD}	10	–	–	ns	–
MPD hold time from $\overline{MPDS}$	t_{hPD}	10	–	–	ns	–
Read Interval	t_{RDI}	13	–	–	ns	–
Read access to write access and write access to read access inactive time	t_{WRRD}	13	–	–	ns	–
MPD delay time from $\overline{MPDTACK}$ active	t_{dD}	-8	–	0	ns	–

7.11.1.3 Synchronous Motorola Interface Timing

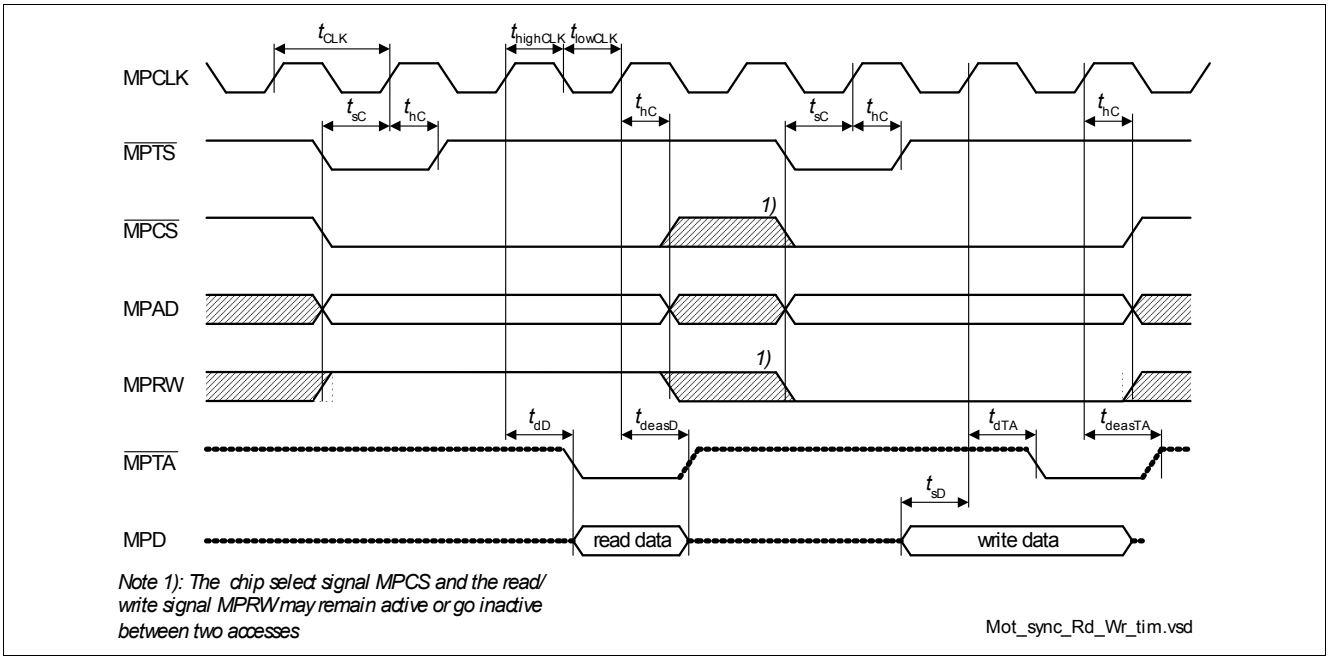


Figure 92 Synchronous Motorola Write/Read Timing

Table 48 Synchronous Motorola Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
MPCLK clock period	t_{CLK}	20	–	–	ns	–
MPCLK clock high time	$t_{highCLK}$	8	–	–	ns	–
MPCLK clock low time	t_{lowCLK}	8	–	–	ns	–
Control ($\overline{MPTS}$, $\overline{MPCS}$, MPAD, MPRW) setup time	t_{sC}	8.25	–	–	ns	–
Control ($\overline{MPTS}$, $\overline{MPCS}$, MPAD, MPRW) hold time	t_{hC}	5	–	–	ns	–
MPD delay time	t_{dD}	1	–	10.25	ns	–
$\overline{MPTA}$ delay time	t_{dTA}	1	–	10.25	ns	–
MPD deasserted to high Z	t_{deasD}	1	–	10.25	ns	–
$\overline{MPTA}$ deasserted to high Z[1)]	t_{deasTA}	1	–	10.25	ns	–
MPD setup time	t_{sD}	8.25	–	–	ns	–

1) To hold this timing requirement pin $\overline{MPTA}$ is driven ONE for 8 ns

7.11.1.4 Serial Control Interface Timing

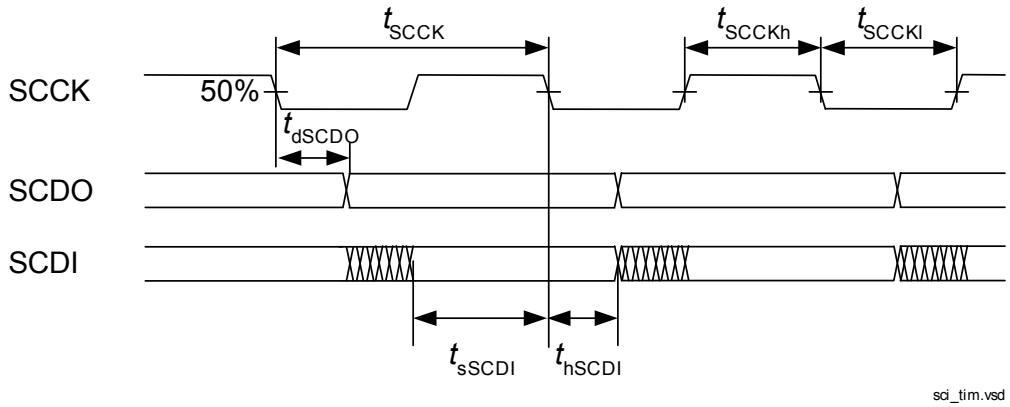


Figure 93 Serial Control Interface Timing

Table 49 Serial Control Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
SCCK clock period[1)]	t_{SCCK}	67		667	ns	–
SCCK clock high time[1)]	t_{SCCKh}	33	[2)]	–	ns	–
SCCK clock low time[1)]	t_{SCCKl}	33	[3)]	–	ns	–
SCDI setup time	t_{sSCDI}	[4)]	–	–	ns	–
SCDI hold time	t_{hSCDI}	0	–	–	ns	–
SCDO delay time	t_{dSCDO}	5	–	13	ns	–

1) SCCK is programmable. Timing is the same for either edge or polarity used.
Clock frequencies in the range from 1.5 to 15 MHz can be applied
2) Typical value for t_{SCCKh} is $t_{SCCK}/2$
3) Typical value for t_{SCCKl} is $t_{SCCK}/2$
4) Minimum value for t_{sSCDI} is t_{SCCKh}

7.11.1.5 Serial Peripheral Interface

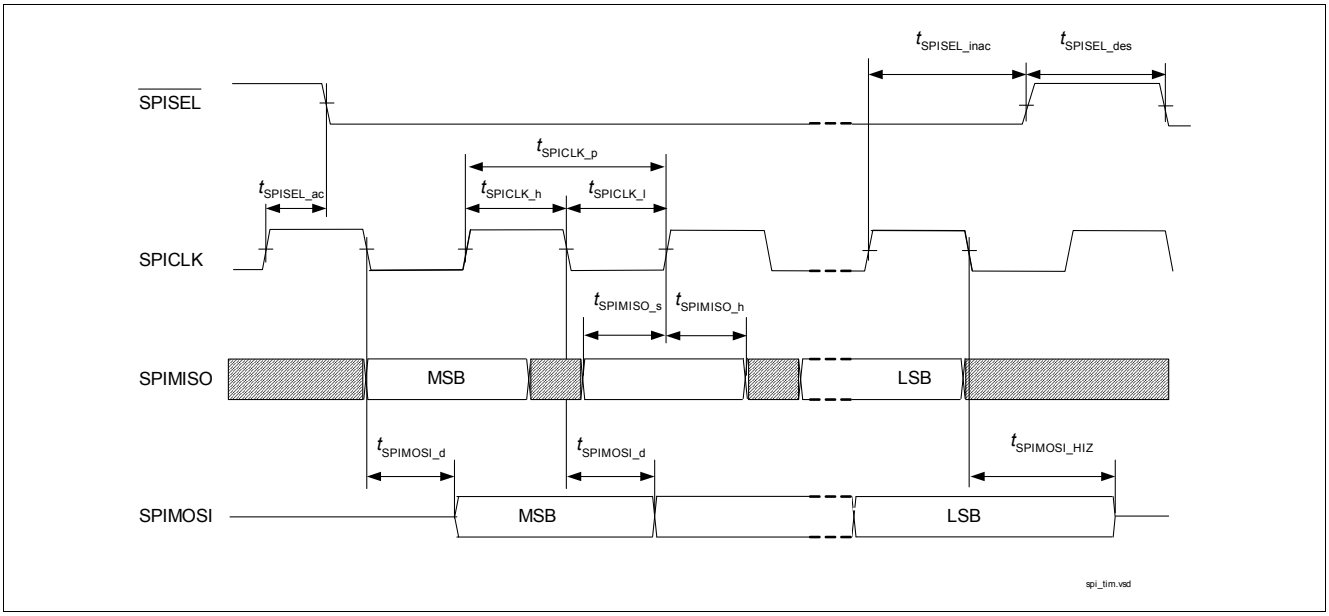


Figure 94 Serial Peripheral Interface Timing

Table 50 Serial Peripheral Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
SPICLK clock period	t_{SPICLK_p}	50	–	–	ns	–
SPICLK clock high	t_{SPICLK_h}	20	–	–	ns	–
SPICLK clock low	t_{SPICLK_l}	20	–	–	ns	–
$\overline{\text{SPISEL}}$ active delay from Clock	t_{SPISEL_ac}	15	–	20	ns	–
$\overline{\text{SPISEL}}$ inactive delay from Clock	t_{SPISEL_inac}	25	–	30	ns	–
$\overline{\text{SPISEL}}$ deselect time	t_{SPISEL_des}	100	–	–	ns	–
SPIMOSI delay from SPICLK	$t_{SPIMOSI\_d}$	0	–	10	ns	–
SPIMOSI deasserted to high Z	$t_{SPIMOSI\_HIZ}$	–	–	20		–
SPIMISO setup time	$t_{SPIMISO\_s}$	0	–	–	ns	–
SPIMISO hold time	$t_{SPIMISO\_h}$	10	–	–	ns	–

7.11.2 System Interface 1

7.11.2.1 MII Interface

This chapter describes the MII timing for the PHY and MAC modes.

For the Virtual PHY mode the timing according to the MAC mode applies.

PHY Mode

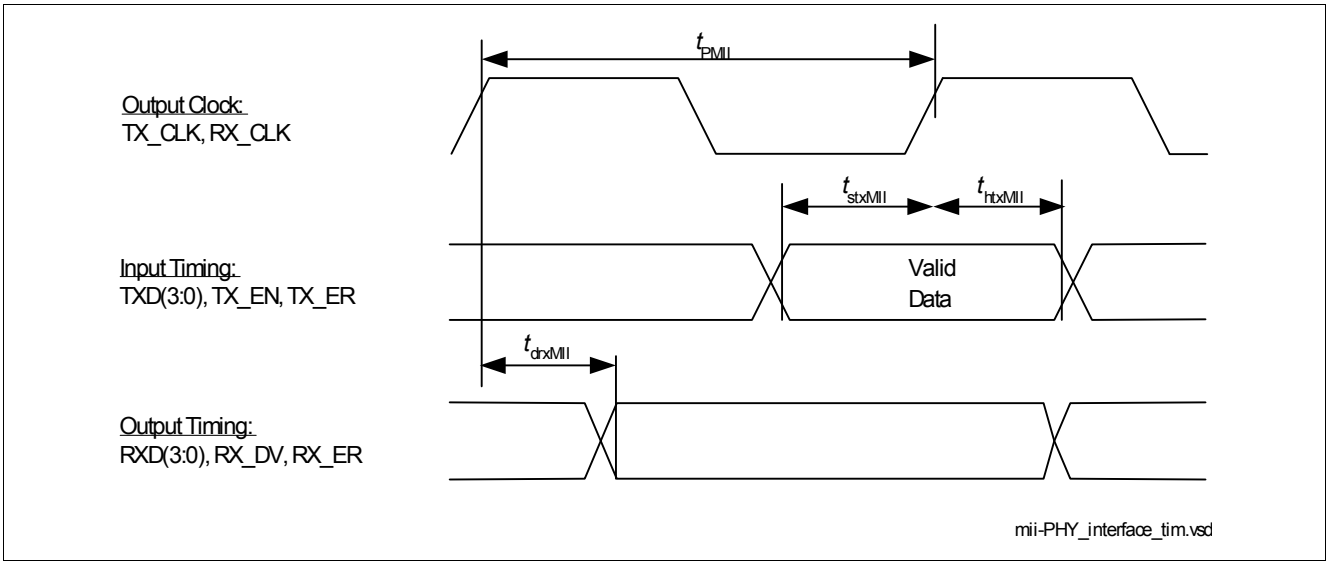


Figure 95 MII Timing, PHY Mode

Table 51 MII Interface Timing, PHY Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
TX_CLK/RX_CLK clock output period 100 Mbit/s	t_{PMII}	24.999	25	25.001	MHz	± 50 ppm[1] (according to standard)
TX_CLK/RX_CLK clock output period 10 Mbit/s	t_{PMII}	2.4999	2.5	2.5001	MHz	± 50 ppm[1] (according to standard)
TX_CLK/RX_CLK clock output duty cycle		35	50	65	%	–
RXD(3:0), RX_DV, RX_ER delay time	t_{drxMII}	15	–	25	ns	–
TXD(3:0), TX_EN, TX_ER setup time	t_{stxMII}	10	–	–	ns	–
TXD(3:0), TX_EN, TX_ER hold time	t_{htxMII}	0	–	–	ns	–

1) Accuracy depends on the connected crystal (see Table 44)

MAC Mode

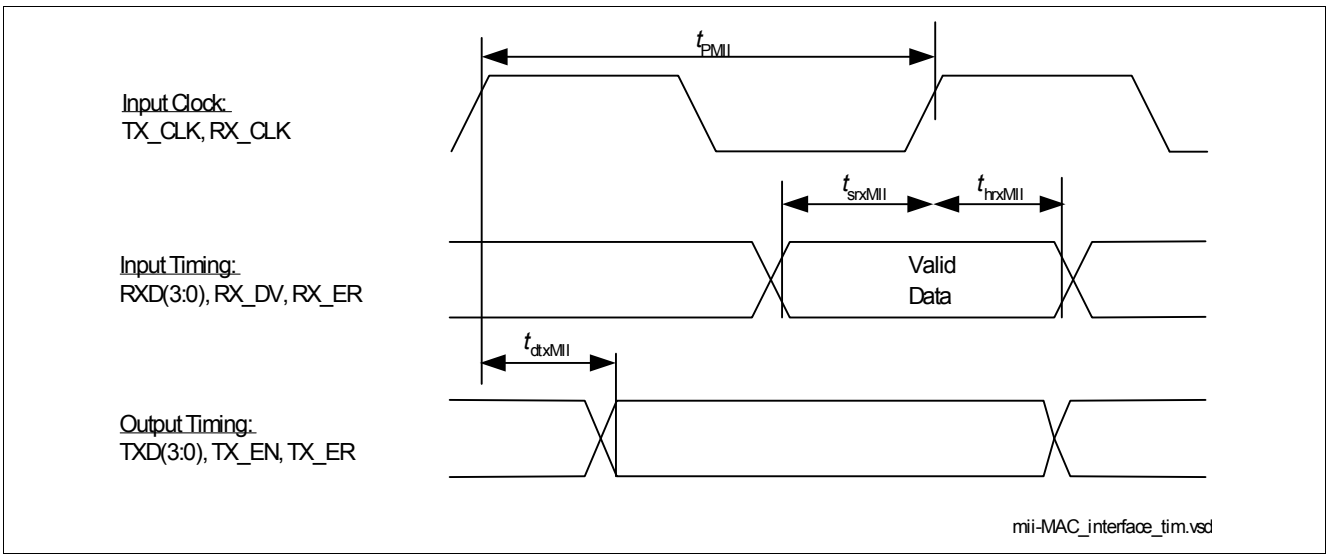


Figure 96 MII Timing, MAC Mode

Table 52 MII Interface Timing, MAC Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
TX_CLK/RX_CLK clock input period 100 Mbit/s	t_{PMII}	24.975	25	25.025	MHz	± 1000 ppm
TX_CLK/RX_CLK clock input period 10 Mbit/s	t_{PMII}	2.4975	2.5	2.5025	MHz	± 1000 ppm
TX_CLK/RX_CLK clock input duty cycle		35	50	65	%	–
TXD(3:0), TX_EN, TX_ER delay time	t_{dtxMII}	0	–	25	ns	(according to standard)
RXD(3:0), RX_DV, RX_ER setup time	t_{srxMII}	10	–	–	ns	(according to standard)
RXD(3:0), RX_DV, RX_ER hold time	t_{hrxMII}	10	–	–	ns	(according to standard)

7.11.2.2 RMII Interface

This chapter describes the RMII interface timing for the PHY and MAC modes.

For the Virtual PHY mode the timing according to the MAC mode applies.

To simplify the board design, the reference output clock pin REF_CLK_o as specified in **Figure 97** and **Table 53** is provided by the SOCRATES™-4e. If the reference output clock pin REF_CLK_o is used it has to be connected to pin REFCLK_i.

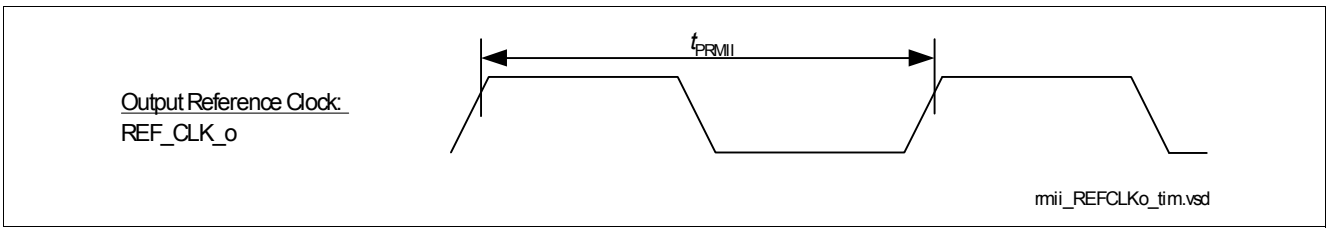


Figure 97 REFCLK_o Timing

Table 53 RMII, Reference Clock Output

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
REF_CLK_o clock output period 10 Mbit/s and 100 Mbit/s	t_{PRMII}	49.99	50	50.003	MHz	± 50 ppm[1] (according to RMII consortium)
REF_CLK_o clock output duty cycle		35	50	65	%	(according to RMII consortium)

1) Accuracy depends on the connected crystal (see **Table 44**)

PHY Mode

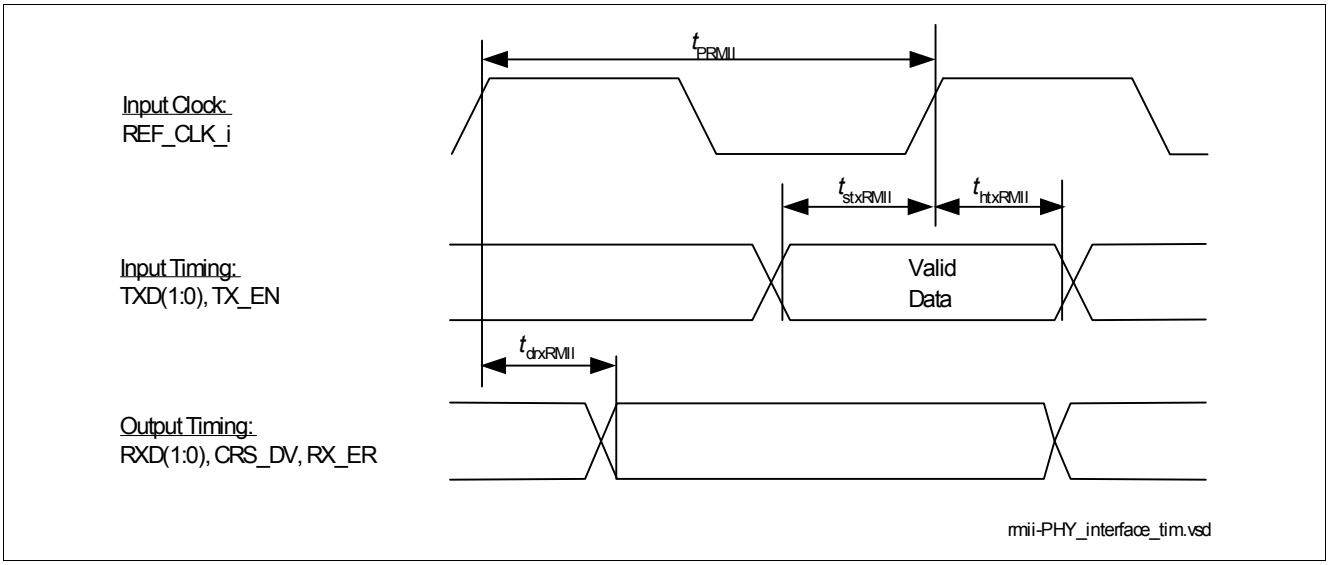


Figure 98 RMII Timing, PHY Mode

Table 54 RMII Interface Timing, PHY Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
REF_CLK_i clock input period 10 Mbit/s and 100 Mbit/s	t_{PRMII}	49.95	50	50.05	MHz	± 1000 ppm
REF_CLK_i clock input duty cycle		35	50	65	%	–
RXD(1:0), CRS_DV, RX_ER delay time	$t_{drxRMII}$	4	–	11	ns	25 pF load
TXD(1:0), TX_EN setup time	$t_{stxRMII}$	4	–	–	ns	(according to RMII consortium)
TXD(1:0), TX_EN hold time	$t_{htxRMII}$	2	–	–	ns	(according to RMII consortium)

MAC Mode

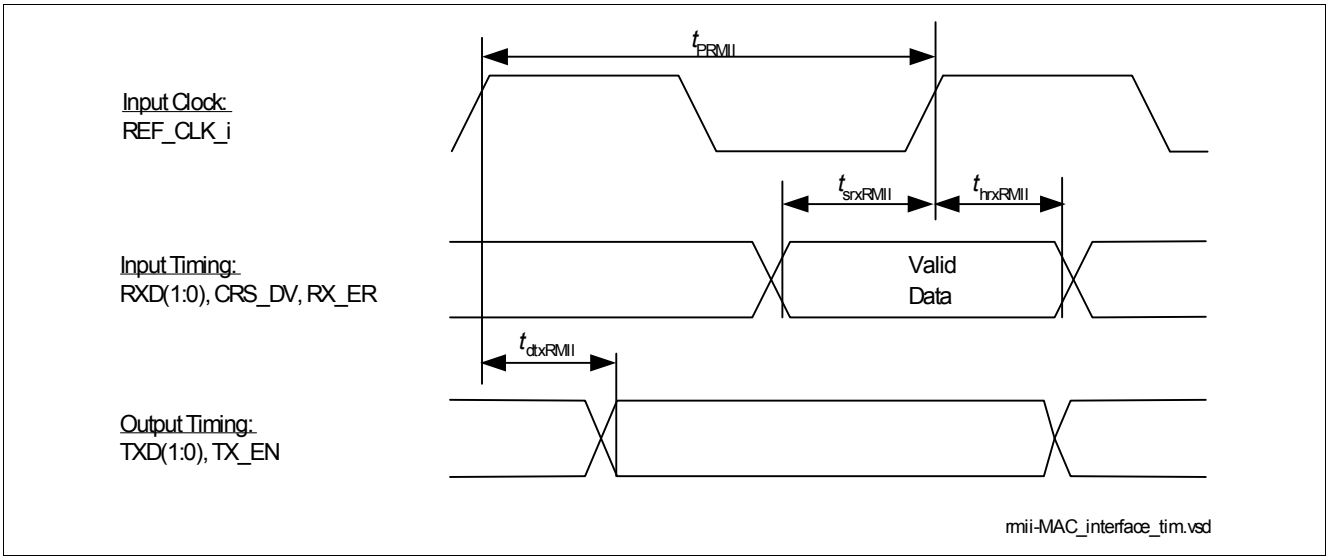


Figure 99 RMII Timing, MAC Mode

Table 55 RMII Interface Timing, MAC Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
REF_CLK_i clock input period 10 Mbit/s and 100 Mbit/s	t_{PRMII}	49.95	50	50.05	MHz	± 1000 ppm
REF_CLK_i clock input duty cycle		35	50	65	%	–
TXD(1:0), TX_EN delay time	$t_{dtxRMII}$	4	–	11	ns	25 pF load
RXD(1:0), CRS_DV, RX_ER setup time	$t_{srxRMII}$	4	–	–	ns	(according to RMII consortium)
RXD(1:0), CRS_DV, RX_ER hold time	$t_{hrxRMII}$	2	–	–	ns	(according to RMII consortium)

7.11.2.3 SS-SMII Interface

This chapter describes the source synchronous SMII (SS-SMII) timing for the PHY and MAC modes.

For the Virtual PHY mode the timing according to the MAC mode applies.

To simplify the board design, the reference output clock pin CLOCK_o as specified in **Figure 100** and **Table 56** is provided by the SOCRATES™-4e. If the reference output clock pin CLOCK_o is used it has to be connected to pin CLOCK_i.

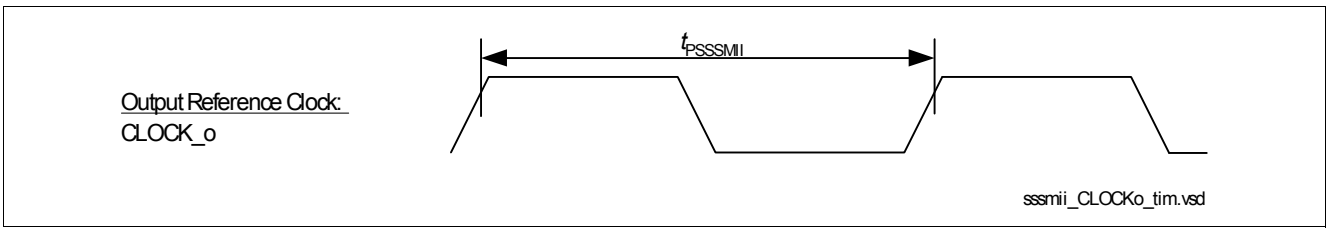


Figure 100 CLOCK_o Timing

Table 56 SS-SMII, Reference Clock Output

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
CLOCK_o clock output period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.99	125	125.01	MHz	± 50 ppm[1)]
CLOCK_o clock output duty cycle		35	50	65	%	–

1) Accuracy depends on the connected crystal (see **Table 44**)

PHY Mode

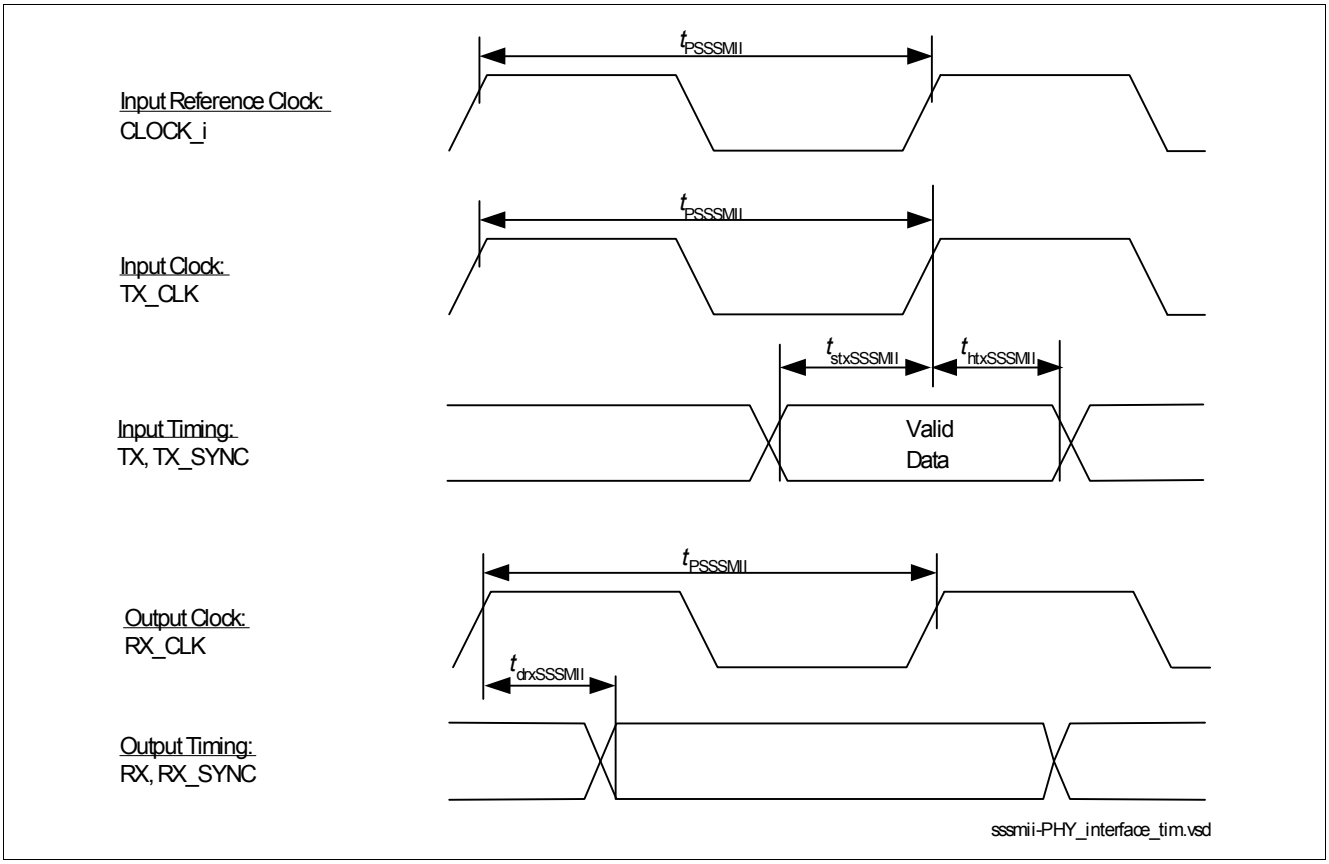


Figure 101 SS-SMII Timing, PHY Mode

Table 57 SS-SMII Interface Timing, PHY Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
CLOCK_i clock input period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.87	125	125.13	MHz	± 1000 ppm
CLOCK_i clock input duty cycle		35	50	65	%	–
TX_CLK clock input period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.87	125	125.13	MHz	± 1000 ppm
TX_CLK clock input duty cycle		35	50	65	%	–
RX_CLK clock output period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.99	125	125.01	MHz	± 50 ppm[1)]
RX_CLK clock output duty cycle		35	50	65	%	–
RX, RX_SYNC delay time	$t_{drxSSSMII}$	1.5	–	5	ns	40 pF load[2)] (according to SMII Spec)
TX, TX_SYNC setup time	$t_{stxSSSMII}$	1	–	–	ns	(1.5 ns according to SMII Spec)
TX, TX_SYNC hold time	$t_{htxSSSMII}$	0.5	–	–	ns	(1 ns according to SMII Spec)

1) Accuracy depends on the connected crystal (see **Table 44**)

2) Same Load for Clock and Data assumed

MAC Mode

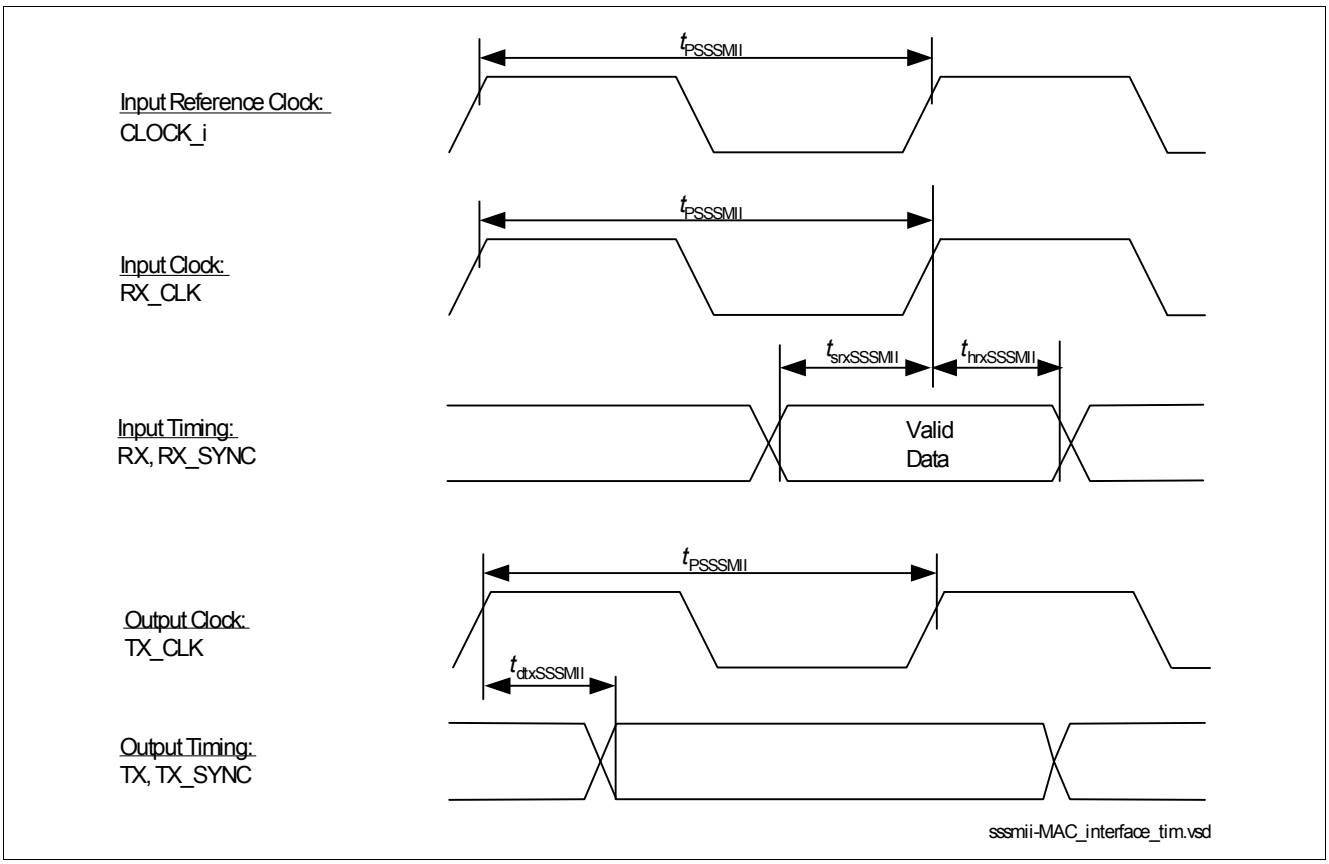


Figure 102 SS-SMII Timing, MAC Mode

Table 58 SS-SMII Interface Timing, MAC Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
CLOCK_i clock input period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.87	125	125.13	MHz	± 1000 ppm
CLOCK_i clock input duty cycle		35	50	65	%	–
RX_CLK clock input period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.87	125	125.13	MHz	± 1000 ppm
RX_CLK clock input duty cycle		35	50	65	%	–
TX_CLK clock output period 10 Mbit/s and 100 Mbit/s	$t_{PSSSMII}$	124.99	125	125.01	MHz	± 50 ppm[1)]
TX_CLK clock output duty cycle		35	50	65	%	–
TX, TX_SYNC delay time	$t_{dtxSSSMII}$	1.5	–	5	ns	40 pF load[2)] (according to SMII Spec)
RX, RX_SYNC setup time	$t_{srxSSSMII}$	1	–	–	ns	(1.5 ns according to SMII Spec)
RX, RX_SYNC hold time	$t_{hrxSSSMII}$	0.5	–	–	ns	(1 ns according to SMII Spec)

1) Accuracy depends on the connected crystal (see **Table 44**)

2) Same Load for Clock and Data assumed

7.11.2.4 MDIO Interface

This chapter describes the MDIO timing for the PHY and MAC modes.

For the Virtual PHY mode the timing according to the PHY mode applies.

PHY Mode

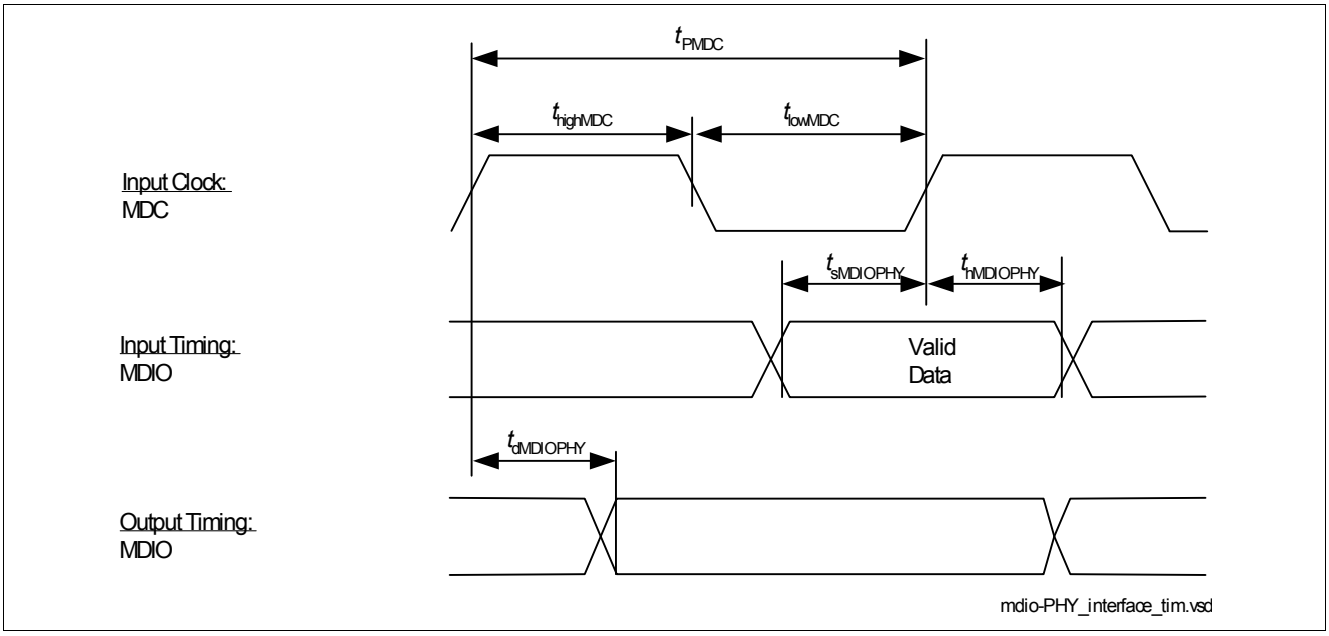


Figure 103 MDIO Timing, PHY Mode

Table 59 MDIO Interface Timing, PHY Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
MDC clock period	t_{PMDIO}	400	–	–	ns	(according to standard)
MDC clock high time	$t_{highMDC}$	160	–	–	ns	(according to standard)
MDC clock low time	t_{lowMDC}	160	–	–	ns	(according to standard)
MDIO delay time, sourced by PHY	$t_{dMDIOPHY}$	10	–	300	ns	(according to standard: 0...300 ns)
MDIO setup time, sourced by MAC	$t_{sMDIOPHY}$	10	–	–	ns	(according to standard)
MDIO hold time, sourced by MAC	$t_{hMDIOPHY}$	10	–	–	ns	(according to standard)

MAC Mode

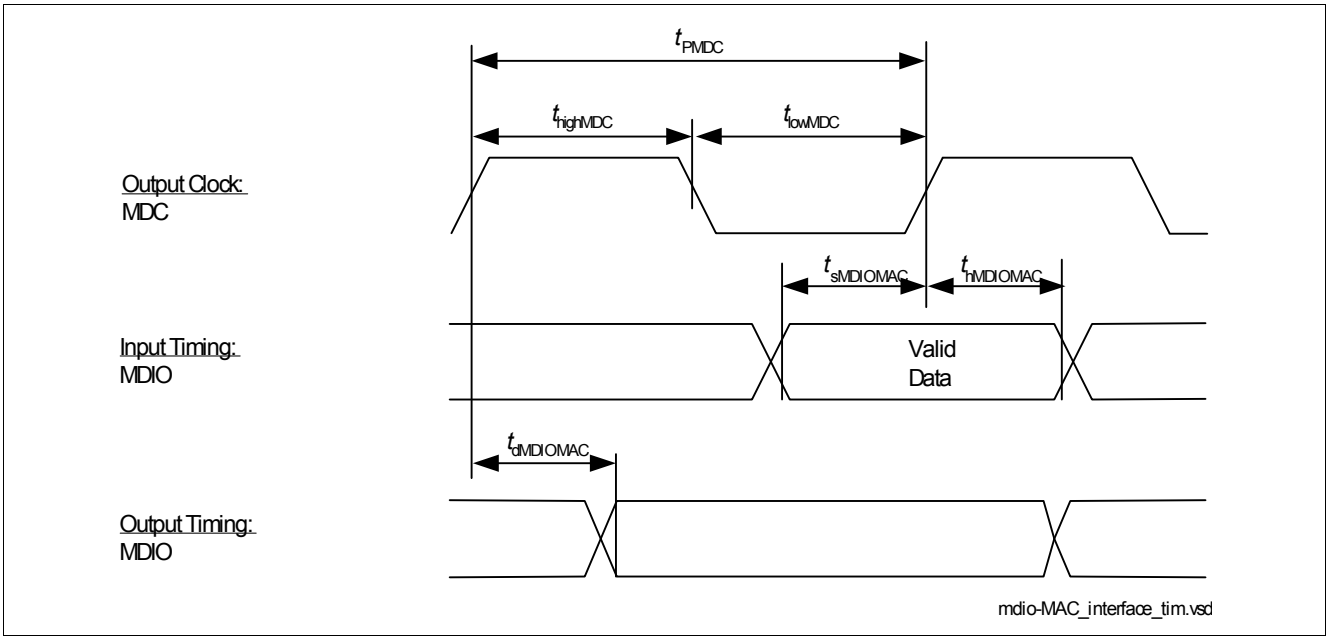


Figure 104 MDIO Timing, MAC Mode

Table 60 MDIO Interface Timing, MAC Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
MDC clock period	t_{PMDIO}	400	–	–	ns	(according to standard)
MDC clock high time	$t_{highMDC}$	160	–	–	ns	(according to standard)
MDC clock low time	t_{lowMDC}	160	–	–	ns	(according to standard)
MDIO delay time, sourced by MAC	$t_{dMDIOMAC}$	20	–	380	ns	–
MDIO setup time, sourced by PHY	$t_{sMDIOMAC}$	40	–	–	ns	–
MDIO hold time, sourced by PHY	$t_{hMDIOMAC}$	0	–	–	ns	–

7.11.2.5 UTOPIA/POSPHY Interface

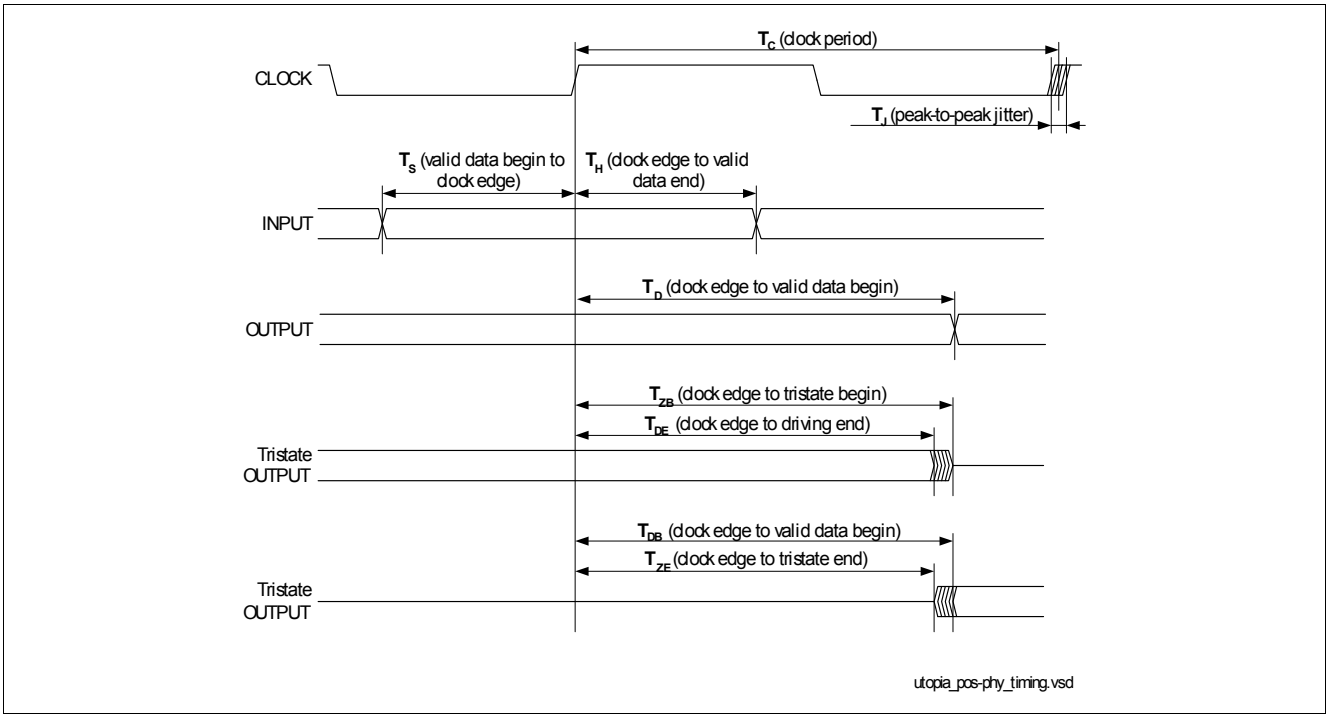


Figure 105 UTOPIA/POS-PHY Timing

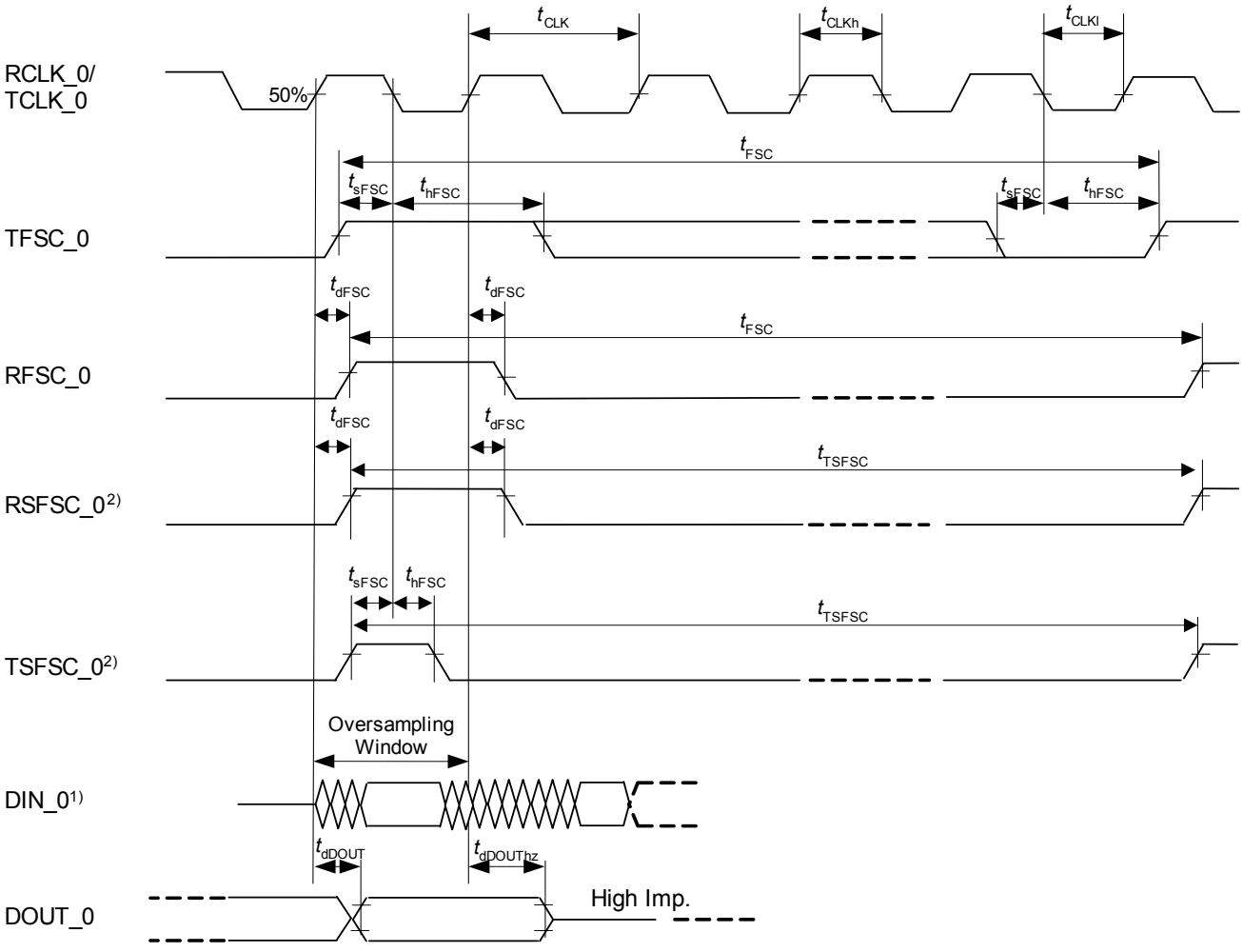
Table 61 UTOPIA/POS-PHY Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
RXCLK, TXCLK clock period	T_C	–	–	50	MHz	–
Clock duty cycle		0.4	–	0.6	Tc	–
Clock rise/fall time		–	–	2	ns	–
Clock peak-to-peak jitter	T_J	–	–	0.005	Tc	–
Setup time	T_S	4/20	–	–	Tc	The UTOPIA L2 standard specifies a minimum value of 4 ns for 50 MHz operation, 8 ns for 33 MHz operation and 10 ns for 25 MHz operation, the setup time for different frequencies is calculated by multiplying the 4/20 with the clock period of the used frequency [Hz]
Hold time	T_H	1	–	–	ns	–

Table 61 UTOPIA/POS-PHY Interface Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Delay time for outputs	T_D	2	–	12/20	Tc	The POS-PHY L2 standard specifies a maximum value of 12 ns for 50 MHz operation, the max. delay time for the outputs for different frequencies is calculated by multiplying the 12/20 with the clock period of the used frequency [Hz]
Delay time to driving end for tristate outputs	T_{DE}	2	–	–	ns	–
Delay time to high Z begin for tristate outputs	T_{ZB}	2	–	12/20	Tc	the max. delay time for the outputs for different frequencies is calculated by multiplying the 12/20 with the clock period of the used frequency [Hz]
Delay time to high Z end for tristate outputs	T_{ZE}	2	–	–	ns	–
Delay time for tristate outputs	T_{DB}	2	–	12/20	Tc	the max. delay time for the outputs for different frequencies is calculated by multiplying the 12/20 with the clock period of the used frequency [Hz]

7.11.3 System Interface 2 (TDM)



1) DIN_# is oversampled with the internal system clock.
The data is evaluated according to majority decision, i.e. the data which is valid for at least 20 ns plus 50% of the oversampling window is sampled.
This means for example for a clock frequency of 16.384 kHz the data delay of the counter part must not exceed 10 ns.

2) The pins TFSC_0 and RFSC_0 can be programmed either as synchronisation pulse (125 µs) or multiframe sychronization pulses (6 ms). So either a synchronization or multframe synchronization is carried over the pins TFSC_0 and RFSC_0. For illustration purposes both mode at the same time are shown.

PCM_TIM.vsd

Figure 106 TDM Timing

Table 62 TDM Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
RCLK_0/TCLK_0 clock period 1)	t_{CLK}	61 2)		5200 3)	ns	–
RCLK_0/TCLK_0 clock high time 1)	t_{CLKh}	24	4)	–	ns	–
RCLK_0/TCLK_0 clock low time 1)	t_{CLKl}	24	5)	–	ns	–
RFSC_0/TFSC_0 clock period	t_{FSC}	–	125	–	µs	–

Table 62 TDM Interface Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
RSFSC_0/TSFSC_0 clock period[6)]	t_{SFSC}	–	6	–	ms	–
RFSC_0/RSFSC_0 delay time	t_{dFSC}	–	–	10	ns	–
TFSC_0TSFSC setup time	t_{sFSC}	16	–	–	ns	–
TFSC_0/TSFSC hold time	t_{hFSC}	16	–	–	ns	–
DOUT_0 delay time	t_{dDOUT}	–	–	10	ns	–
DOUT_0 delay time to high Z	$t_{dDOUThz}$	–	–	20	ns	–

1) RCLK_0/TCLK_0 is programmable. Timing is the same for either edge or polarity used.
2) In case of selecting 16.384 MHz as TDM data clock
3) In case of selecting 192 kHz as TDM data clock
4) Typical value for t_{CLKh} is $t_{CLK}/2$
5) Typical value for t_{CLKl} is $t_{CLK}/2$
6) Multiframe information needs to be enabled and is available on pins RFSC_0/TFSC_0

7.11.3.1 JTAG Interface

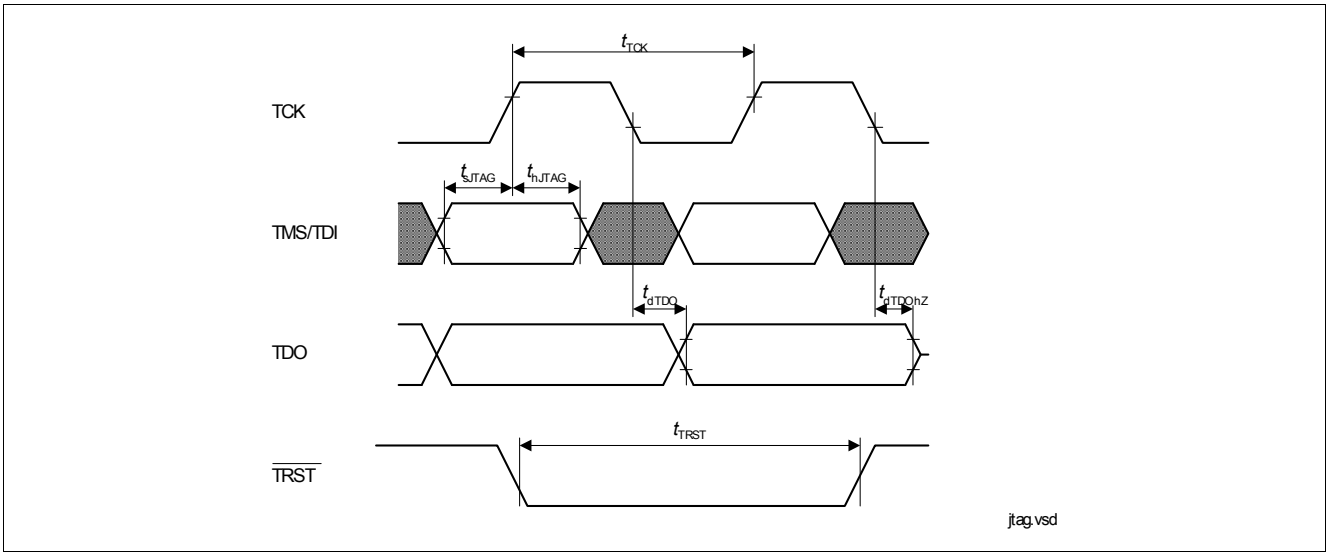


Figure 107 JTAG Interface Timing

Table 63 JTAG Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
TCK clock period	t_{TCK}	160	–	–	ns	–
TCK clock duty cycle		35	50	65	%	–
TDO delay time	t_{dTDO}	0	–	30	ns	–
TDO delay time to high Z	t_{dTDOhz}	0	–	30	ns	–
TMS, TDI setup time	t_{sJTAG}	10	–	–	ns	–
TMS, TDI hold time	t_{hJTAG}	10	–	–	ns	–
$\overline{\text{TRST}}$ pulse width	t_{TRST}	1)	–	–	ns	–

1) Typical value for t_{TRST} is 2*t_{TCK}

8 Package Outline

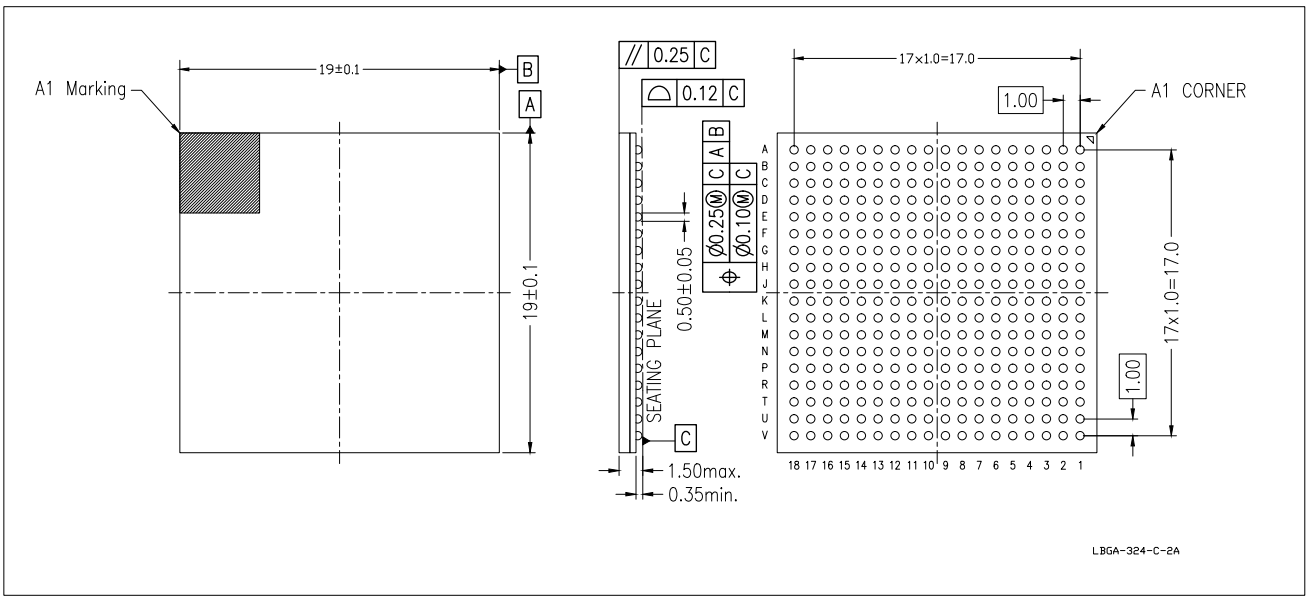


Figure 108 PG-LBGA-324-8 (Plastic Low Profile Ball Grid Array Package)

Package description, package handling, PCB and board assembly information is available on request.

Table 64 Thermal Resistance

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Thermal Resistance Top	$R_{th\text{-}JCtop}$	–	6.0	–	K/W	Top of package
Thermal Resistance Bottom	$R_{th\text{-}JCbottom}$	–	3.6	–	K/W	Bottom of package

Terminology

A	
ADC	Analog-Digital-Converter
AGC	Automatic Gain Control
AFE	Analog Front End
AIS	Alarm Indication Signal
ATM	Asynchronous Transfer Mode
AAL	ATM Adaption Layer
B	
BLB	Backward Loop Back
C	
CC	Continuity Check
CO	Central Office
CPE	Customer Premises Equipment
D	
DAC	Digital-Analog-Converter
DCP	Device Control Protocol
DEVAD	Device Address
DFE	Digital Front End
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DTE	Data Terminal Equipment
E	
EFM	Ethernet First Mile according to IEEE 802.3-2004
F	
FLB	Forward Loop Back
H	
HDLC	High-level Data Link Control
Host I/F	Host Interface stands for microprocessor interface (Motorola synchronous, Motorola asynchronous, Intel demux), serial control interface (SCI) and MDIO
I	
IDC	Integrated Device Controller
I/F	Interface
L	
LOC	Loss of Connection
M	
MAC	Medium Access Control
MAC(TC)	MAC of TC. Stands for MAC of EFM-, HDLC- or ATM-TC
MAC(Sys1)	MAC of System Interface 1. Stands for MAC of xMII, UTOPIA or POS-PHY

MAU	Medium Attachment Unit
MDIO	Management Data Input/Output
MMD	MDIO Manageable Device
MIB	Management Information Base
MII	Media Independent Interface
N	
NWP	Network Processor
O	
OAM	Operation And Maintenance
OP	Operation Code
P	
PAF	PME Aggregation Function according to IEEE 802.3-2004
PCS	Physical Coding Sublayer according to IEEE 802.3-2004
PHY	Physical Layer Device
PHYAD	PHY Address
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent (device)
PME	Physical Medium Entity
POS-PHY	Packet Over SONET PHY interface
ppm	Parts per million
PRTAD	Port Address
PRE	Preamble
PSE	Power Sourcing Equipment
PTM	Packet Transfer Mode
R	
RDI	Remote Defect Indication
REGAD	Register Address
RFC	Request For Comment
RMII	Reduced MII
S	
Serial TC	All TCs standardized in ITU-T G.991.2 (2004) except ATM-TC and PTM-TC
SRU	SHDSL Regenerator Unit
SRU-C	SHDSL Regenerator Unit COT side
SRU-R	SHDSL Regenerator Unit RT side
SS-SMII	Source Synchronous Serial MII
ST	Start of frame
STA	Station Management
STU	SHDSL Terminal Unit
STU-C	SHDSL Terminal Unit COT side
STU-R	SHDSL Terminal Unit RT side

T

TA	Turnaround
TC	Transmission Convergence
TPS-TC	Transmission Protocol Specific TC

U

UTOPIA	Universal Test and Operations Physical Interface for ATM

V

VLAN	Virtual Local Area Network

X

xDSL	“Any” DSL, (ADSL, SHDSL or VDSL)
xMII	“Any” MII (MII, SS-SMII and RMII)
xTC	“Any” TPS-TC, (EFM, Packet and ATM)

www.lantiq.com

Published by Lantiq Beteiligungs-GmbH & Co.KG