4538 PMC® T1/E1/J1 Communications Controller Hardware Reference Manual

Document No. UG4538-001-A

Print Date: October 2001

Copyright Notice

Printed in the United States of America, 2001.

Interphase Corporation
13800 Senlac
Dallas, Texas 75234

Phone: (214) 654-5000
Fax: (214) 654-5506

Disclaimer

Information in this manual supersedes any preliminary specifications, preliminary data sheets, and prior versions of this manual. While every effort has been made to ensure the accuracy of this manual, Interphase Corporation assumes no liability resulting from omissions, or from the use of information obtained from this manual. Interphase Corporation reserves the right to revise this manual without obligation to notify any person of such revision. Information available after the printing of this manual will be in one or more Read Me First documents. Each product shipment includes all current Read Me First documents. All current Read Me First documents are also available on our web site.

THIS MANUAL IS PROVIDED "AS IS." INTERPHASE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL INTERPHASE BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Trademark Acknowledgments

Assistance

Product Purchased from Reseller

Contact the reseller or distributor if

- You need ordering, service or any technical assistance.
- You received a damaged, incomplete or incorrect product.

Product Purchased Directly from Interphase Corporation

Contact Interphase Corporation directly for assistance with this, or any other Interphase Corporation product. Please have your purchase order and serial numbers ready.

Customer Support

United States:	Telephone:	(214) 654-5555
	Fax:	(214) 654-5506
	E-Mail:	intouch@iphase.com
Europe:	Telephone:	+ 33 (0) 1 41 15 44 00
	Fax:	+ 33 (0) 1 41 15 12 13

World Wide Web

http://www.iphase.com

Anonymous FTP Server

ftp.iphase.com

END-USER LICENSE AGREEMENT
FOR INTERPHASE CORPORATION SOFTWARE

IMPORTANT NOTICE TO USER–READ CAREFULLY

THIS END-USER LICENSE AGREEMENT FOR INTERPHASE CORPORATION SOFTWARE ("AGREEMENT") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR SINGLE ENTITY) AND INTERPHASE CORPORATION FOR THE SOFTWARE PRODUCTS ENCLOSED HEREIN WHICH INCLUDES COMPUTER SOFTWARE AND PRINTED MATERIALS ("SOFTWARE"). BY INSTALLING, COPYING, OR OTHERWISE USING THE ENCLOSED SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY RETURN, WITHIN THIRTY DAYS, THE UNUSED SOFTWARE TO THE PLACE FROM WHICH YOU OBTAINED IT FOR A FULL REFUND.

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold.

Grant of License: You are granted a personal license to install and use the Software on a single computer solely for internal use and to make one copy of the Software in machine readable form solely for backup purposes.

Restrictions on Use: You may not reverse engineer, decompile, or disassemble the Software. You may not distribute copies of the Software to others or electronically transfer the Software from one computer to another over a network. You may not use the Software from multiple locations of a multi-user or networked system at any time. You may not use this software on any product for which it was not intended. You may not use this software on any non-Interphase product. LICENSEE MAY NOT RENT, LEASE, LOAN, OR RESELL THE SOFTWARE OR ANY PART THEREOF.

Ownership of Software: Interphase or its vendors retain all title to the Software, and all copies thereof, and no title to the Software, or any intellectual property in the Software, is being transferred.

Software Transfer: You may permanently transfer all of your rights under this Agreement, provided you retain no copies, you transfer all the Software, and the recipient agrees to the terms of this Agreement.

Limited Warranty: Interphase Corporation ("Seller") warrants that (i) the hardware provided to Buyer ("Products") shall, at the F.O.B. point, be free from defects in materials and workmanship for a period of one (1) year from the date of shipment to Buyer; (ii) the software and/or firmware associated with or embedded in the Products shall comply with the applicable specifications for a period of six (6) months from the date of shipment to Buyer; and (iii) its services will, when performed, be of good quality. Defective and nonconforming Products and software must be held for Seller's inspection and returned at Seller's request, freight prepaid, to the original F.O.B. point.

Upon Buyer's submission of a claim in accordance with Seller's Return and Repair Policy, Seller will, at its option either (i) repair or replace the nonconforming Product; (ii) correct or replace the software/firmware; (iii) rework the nonconforming services; or (iv) refund an equitable portion of the purchase price attributable to such nonconforming Products, software, or services. Seller shall not be liable for the cost of removal or installation of products or any unauthorized warranty work, nor shall Seller be responsible for any transportation costs, unless expressly authorized in writing by Seller. This warranty does not cover damage to the Product resulting from accident, disaster, misuse, negligence, improper maintenance, or modification or repair of the Product other than by Seller. Any Products or software replaced by Seller will become the property of Seller.

REMEDIES AND EXCLUSIONS. THE SOLE LIABILITY OF SELLER AND BUYER'S SOLE REMEDY FOR BREACH OF THESE WARRANTIES SHALL BE LIMITED TO REPAIR OR REPLACEMENT OF THE PRODUCTS OR CORRECTION OF THAT PART OF THE SOFTWARE, WHICH FAILS TO CONFORM TO THESE WARRANTIES. EXCEPT AS EXPRESSLY STATED HEREIN, AND EXCEPT AS TO TITLE, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, IN CONNECTION WITH OR ARISING OUT OF ANY PRODUCT OR SOFTWARE PROVIDED TO BUYER.

IN NO EVENT SHALL SELLER HAVE ANY LIABILITY FOR INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THESE WARRANTIES, INCLUDING BUT NOT LIMITED TO LOSS OF ANTICIPATED PROFITS, LOSS OF DATA, USE OR GOODWILL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. (IC-199, 1/97)

Limitation of Liability: NEITHER INTERPHASE NOR ITS LICENSORS SHALL BE LIABLE FOR ANY GENERAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR OTHER DAMAGES ARISING OUT OF THIS AGREEMENT EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Confidentiality: The Software is copyrighted and contains proprietary and confidential trade secret information of Interphase and its vendors. Licensee agrees to maintain the Software in confidence and not to disclose the Software to any third party without the express written consent of Interphase. Licensee further agrees to take all reasonable precautions to prevent access to the Software by unauthorized persons.

Termination: Without prejudice to any other rights, Interphase may terminate this Agreement if you fail to comply with any term or condition of the Agreement. In such event you must destroy the Software together with all copies, updates, or modifications thereof.

Export: You agree to comply with all export and re-export restrictions and regulations of the U.S. Department of Commerce or other applicable U.S. agency. You must not transfer the Software to a prohibited country or otherwise violate any such restrictions or regulations.

U.S. Government Restricted Rights: Use, duplication, or disclosure of the Software to or by the U.S. Government is subject to restrictions as set forth in the applicable U.S. federal procurement regulations covering commercial/restricted rights software. You are responsible for complying with the notice requirements contained in such regulations.

General: You acknowledge that you have read and understand this Agreement, and by installing and using the Software you agree to be bound by the terms and conditions herein. You further agree that this is the complete and exclusive Agreement between Interphase and yourself. No variation of the terms of this Agreement or any different terms will be enforceable against Interphase unless agreed to in writing by Interphase and yourself. The validity of this Agreement and the rights, obligations, and relations of the parties hereunder shall be determined under the substantive laws of the State of Texas. If any provision of this Agreement is held invalid, illegal, or unenforceable, the remaining provisions shall in no way be affected or impaired thereby. All rights in the Software not specifically granted in this Agreement are reserved by Interphase.

Contents

List of Figures

List of Tables

List of Examples

Using This Guide

Purpose

This *4538 Hardware Reference Manual* is designed for software developers in Interphase customer organizations who intend to develop embedded software and/or host drivers for the 4538 T1/E1/J1 communications controller.

The 4538 is delivered with an Interphase Boot Firmware located in the FLASH memory. This firmware initializes and configures the 4538 hardware at each boot. It also includes built-in self tests and a monitor. Software developers can decide to keep the Interphase Boot Firmware and develop an operational firmware that will start after the completion of this Interphase Boot Firmware. This solution is recommended for the following reasons:

- PowerQUICC II and 4538 initial hardware configuration code is already developed and validated by Interphase in the boot firmware.
- Interphase Boot Firmware provides several ways to download and execute developer's operational firmware.
- Interphase Boot Firmware can be used during the life of the product for operational firmware updates and field unit tests.

However, if the software developers decide to re-develop their own Boot Firmware, this manual describes in detail the 4538 Hardware and provides information relative to its initialization and configuration.

NOTES

To install the 4538 Communications Controller in your CompactPCI Machine, please refer to the *4538 Board Installation and Maintenance Guide*. Different cautions and warnings are described to avoid damage to your communications controller.

All the booting process and software elements composing the Interphase Boot Firmware are described in the *4538 Built-In Self test and Monitor Users Guide*. Refer to this document when using Boot Firmware.

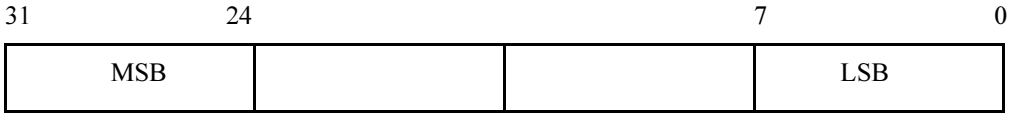
Audience

This guide was written assuming that readers have extensive knowledge of the C programming language and of methods for developing and installing software drivers.

Byte Ordering and Bit Coding Convention

The PCI bus uses the Little Endian Byte ordering: byte 0 in a 32-bit word is the Least Significant Byte (LSB) from an arithmetic point of view and is noted D(7:0). The PowerPC architecture uses the Big Endian Byte ordering: byte 0 in a 32-bit word is the Most Significant Byte (MSB) from an arithmetic point of view and is noted D(31:24).

The PowerPC architecture uses the very unusual Little Endian Bit convention, where bit 0 is on the left and is the most significant bit. Unless otherwise noted, this document does not use this convention. Instead, it uses the classical bit coding convention, where bit 0 (on the right) is the least significant bit and bit i is the 2^i weight bit. This is the Big Endian Bit convention. This coding convention applies to data, addresses, and bit fields. In the following figure, MSB means Most Significant Byte and LSB Least Significant Byte:



The standard C convention is used to identify the numeric format of arithmetical values:

- No prefix for decimal values
- 0x prefix for a hexadecimal value

For example 0x12 = 18.

Type Definition

Only a few basic types are used:

- byte: unsigned, coded as 8 bits
- word: unsigned, coded as two contiguous bytes, most significant first
- dword: unsigned, coded as two contiguous words, most significant first

Code Examples

This document provides several algorithm descriptions presented in PowerPC assembly language and in C language.

Icon Conventions

Icons draw your attention to especially important information:

NOTE

The Note icon indicates important points of interest related to the current subject.

CAUTION

The Caution icon brings to your attention those items or steps that, if not properly followed, could cause problems in your machine's configuration or operating system.

WARNING

The Warning icon alerts you to steps or procedures that could be hazardous to your health, cause permanent damage to the equipment, or impose unpredictable results on the surrounding environment.

Text Conventions

The following conventions are used in this manual. Computer-generated text is shown in typewriter font. Examples of computer-generated text are: program output (such as the screen display during the software installation procedure), commands, directory names, file names, variables, prompts, and sections of program code.

```
Computer-generated text example
```

Commands to be entered by the user are printed in **`bold Courier`** type. For example:

```
cd /usr/tmp
```

Pressing the return key (↵ **`Return`**) at the end of the command line entry is assumed, when not explicitly shown. For example:

```
/bin/su
```

is the same as:

```
/bin/su  ↵ Return
```

Required user input, when mixed with program output, is printed in **`bold Courier`** type. References to UNIX programs and manual page entries follow the standard UNIX conventions.

When a user command, system prompt, or system response is too long to fit on a single line, it will be shown as

```
Do you want the new kernel moved into
\vmunix?[y]
```

with a backslash at either the beginning of the continued line or at the end of the previous line.

Checking and Downloading from the Interphase WWW/FTP Site

The latest production software drivers, firmware, and documentation (in Adobe Acrobat PDF or text format) for our current products are available on our WWW / FTP site. Interphase recommends our customers visit the web site often to verify that they have the latest version of driver, firmware, or documentation.

WWW Method

1. Access the following web page:

   http://www.iphase.com

2. Move the mouse (or other pointer) and click on the `Products` option. A menu will appear on the left side with Telecom and Server I/O options. Choose the appropriate menu item.

3. A new web page with a list of the currently offered products will appear. Choose the correct communications controller for your system bus and configuration needs by clicking on the product number (i.e. 4531S, 6535, 4575, etc.).

4. The Product Description page appears for that product number. At the left side of the page is a list showing further information web pages for that product. Choose the SW Downloads item.

5. A new web page appears with a list of the latest released drivers available for that adapter based on the operating system. Click on the line that describes your driver requirement. Depending on your browser configuration, the driver will now download to your system. If this doesn't work correctly, try to 'right-click' on the proper driver and choose an option that will save the file to your local file system.

FTP Method

1. From your command line, enter following:

   ftp://ftp.iphase.com

2. At the Login: prompt, enter **anonymous**

3. At the Password: prompt, enter your E-mail address.

4. At the ftp prompt, enter **binary**

5. Enter **cd pub**

6. To list all the available product technology directories, enter **dir**. The `00README.txt` file in each directory gives a description of the files and subdirectories in that directory.

7. To access the directory for the technology that you require, enter **cd** followed by the appropriate directory name. Most technology directories also have bus and

operating system subdirectories. In these cases, you must choose the proper bus and operating system by typing **cd <directory>** for the appropriate subdirectories.

8. To download one or more files to your local directory, enter **get <filename>**

9. To exit the FTP site, enter **quit** or **bye**

Hardware Description 1

Overview

The Interphase 4538 PMC E1/T1/J1 Communications Controller is a network interface PCI Mezzanine Card (PMC) equipped with four software-selectable T1/E1/J1 interfaces (two are provided on the front panel). The 4538 board is intended for 2G and 3G wireless networks, Internet access, and Advanced Intelligent Network (AIN) applications.

This chapter provides the functional specification of the 4538. It describes how the different main components of the board are arranged together.

The main components of the 4538 are:

- The PowerQUICC II™, a Motorola® MPC8260 RISC embedded processor.
- The Tundra PowerSpan™, a dedicated PCI bridge that controls the interface between the card and the host 32-bit PCI bus.
- 4 MB of 8-bit FLASH EEPROM memory.
- 32 MB or 64 MB of 64-Bit SDRAM system memory
- The INFINEON QuadFALC™ framers, included in the 4538 communications controller, which control four independent T1/E1/J1 interfaces. For each interface, the QuadFALC includes a framer and a Line Interface Unit (LIU) with data and clock recovery.
- The INTEL LXT971A, an IEEE compliant Fast Ethernet transceiver that supports 10BaseT/100BaseTX auto-negotiation and parallel detection.

4538 Hardware Structure

Figure 1-1 shows the 4538 hardware structure:

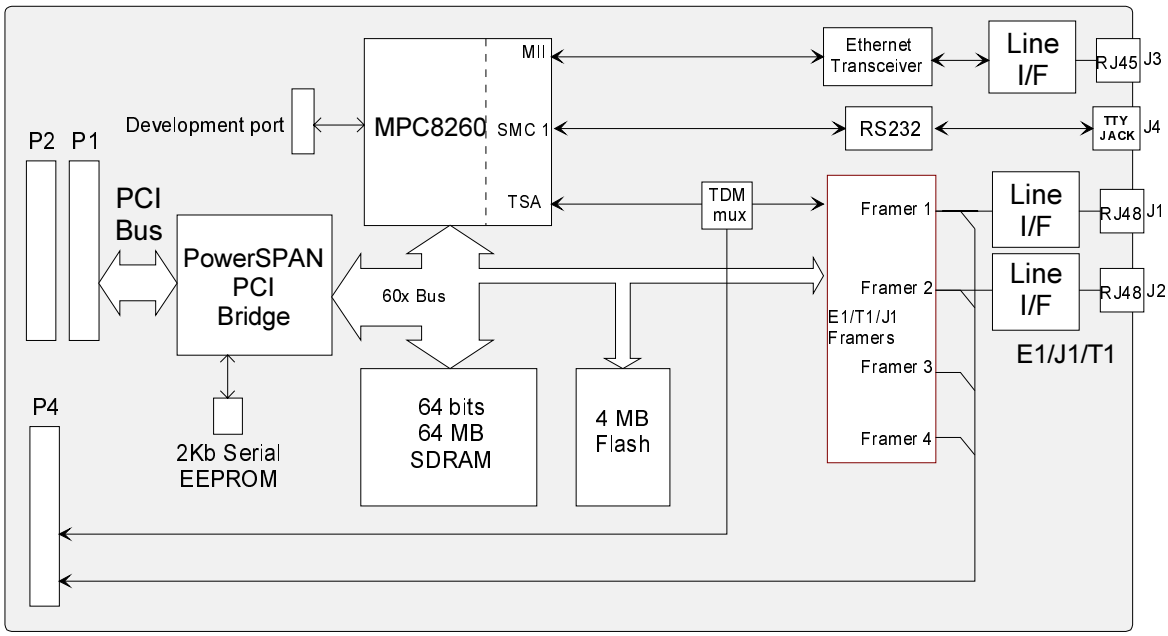


Figure 1-1. 4538 Structure

The PowerQUICC II

The local CPU is a Motorola MPC8260 RISC embedded processor. The MPC8260 includes three major parts:

- An MPC603e core
- A System Interface Unit (SIU)
- A Communication Processor Module (CPM)

The MPC603e core is derived from the PowerPC™ 603e core and includes mainly the integer core and the 16 KB data and 16 KB instruction caches.

The SIU includes a memory management unit and enables control of the external 60x local bus (64-bit data width). The SIU also provides a local bus (32-bit data, 32-bit internal/18-bit external address) used to enhance the operation of the Fast Communication Controllers (FCCs). It can be used to store connection tables for ATM, buffer descriptors, or raw data that is transmitted between channels. It is synchronized with the 60x bus and runs at the same frequency. The 4538 does not provide CPM local memories.

The Communication Processor Module (CPM) is a super-set of the PowerQUICC II CPM with additional capabilities. It features:

- Two Multichannel Communications Controllers (MCCs)

- Three Fast Serial Communications Controllers (FCCs). One is used to control the Ethernet Media-Independent Interface (MII).
- Four Serial Communication Controllers (SCCs)
- Two Serial Management Controllers (SMCs)
- A debug serial port
- A Serial Peripheral Interface (SPI)
- Four timers and an interrupt controller

PowerQUICC II Resets

Once the card is powered-up and the power stabilized, the PowerQUICC II enters into a sequence where it will define certain vital parameters, such as the type of its bus and the PLL multiplication factors. Then it will wait for various conditions, such as PLL stabilization and PCI reset signal de-asserted, before booting.

The PowerQUICC II is controlled by three reset signals:

- –PORESET: Power-on reset
- –HRESET: Hardware reset
- –SRESET: Software Reset

When –PORESET is activated, this also activates –HRESET and –SRESET. –PORESET is the strongest reset. When –HRESET is activated, this also activates –SRESET. When –SRESET is activated, it does not interfere with the other resets (–SRESET is the weakest reset).

A power supervisor controls the MPC8260 input signal –PORESET. It activates –PORESET (0) when the power is not stabilized (at power-up or during power failures). The –PORESET is maintained active for 150 ms after stabilization of the power.

After –PORESET is de-asserted (set to 1), the MPC8260 waits 1024 input clock cycles and samples the MODCK[1:3] bits, which define the default clock multiplication factor and input clock used for the SPLL. The MPC8260 starts its PLL at this time. It maintains –HRESET and –SRESET asserted while the PLL is not locked.

Through its pin –PB_RESET, the PowerSpan also maintains –HRESET asserted as long as the PCI reset signal is activated.

PowerQUICC II –RSTCONF pin is tied to ground, indicating that the MPC8260 is the configuration master. At the rising edge of –HRESET, the MPC8260 generates 64-bit reads into its boot memory (the FLASH) with address starting at 0 and incrimented by 8. The first eight bytes set its Hard Reset Configuration (for detailed initialization see *PowerSpan Hardware Configuration Word* on page 59).

The PowerSpan has no dedicated pin to control the PowerQUICC II hard reset signal –HRESET and soft reset signal –SRESET. Instead, two of its interrupt pins, –INT2 and –INT3 respectively, configured as an output are used. These interrupt are controlled with doorbell bits (see *Hardware and Software Resets Through the PowerSpan* on page 21).

Once all the resets are de-asserted, the PowerQUICC II boots using its 8-bit FLASH device.

The MPC8260 can control the reset of the various communication peripherals through certain CPM I/O ports. When the PowerQUICC II is in reset state, and until it configures these I/O ports as outputs, these reset signals are activated.

System Clocks

The MPC8260 gets its reference clock in its CLOCKIN input pin from a 65.536 MHz reference oscillator.

The MPC8260 input pins MODCK[1:3], along with the MODCK_H field from the Reset Configuration Word define the input clock used for the SPLL and the default clock multiplication factors. The resulting internal system frequencies are:

- PowerPC core frequency: 196.608 MHz (x3)
- CPM frequency: 131.072 MHz (x2)

PCI Local Space Mapping

The PowerPC local processor can address a 4 GB logical space. In this space, the following elements are mapped:

- The vector table (including the reset entry point)
- The MPC8260 internal registers
- The main SDRAM memory
- The FLASH memory
- The PCI bridge (the PowerSpan) and its local-to-PCI window(s)
- The communication peripheral (QuadFALC)

When the MPC8260 boots, it is configured to select the FLASH memory, regardless of the address generated. This will allow the PowerPC to always find the boot start entry in the FLASH. After having booted, having executed a proper jump, and initialized the memory controllers, both the vector table address and the FLASH address can be configured and mapped in other areas: the developer will typically prefer to implement the vector table in a R/W memory device (the main memory SDRAM).

The MPC8260 includes 12 banks with their respective Chip Selects.

The memory mapping has been defined in a way that allows use of the MMU Block Address Translation (BAT) mechanism, which is simpler than the segments-and-pages mechanism. This mechanism divides the memory into several areas that have their own cache properties.

Depending on the device selected, the corresponding memory area can be defined as “cachable” for better performance or must be set as “non cachable”. For instance, the FLASH memory can be cachable. The areas in the SDRAM that are only accessed by the

local processor can also be cachable. The peripherals cannot be cachable. The area of SDRAM memory used for the transfer of data cannot be cachable either, because it can be modified by elements other than the PowerQUICC II, such as the PowerSpan DMA.

In order to simultaneously support cachable and non cachable areas in the SDRAM memories, they are mapped twice in the local space. One mapping area will be defined as cachable and the other will be defined as non cachable.

Table 1-1 and Figure 1-2 indicate the organization of the local space as defined in the current 4538 Boot Firmware code, with the instruction and data BAT blocks and CS banks used.

Table 1-1. PCI Local Space Mapping

Address Area	Size	Element Accessed	IBAT/ DBAT	CS Bank	Property
0x0000 0000 – 0x03FF FFFF	64MB	60x bus Main memory	0/0	1	Cachable
0x8000 0000 – 0x83FF FFFF	64MB	60x bus Main memory (duplicated)	–/1	1	Not cachable
0xC000 0000 – 0xCFFF FFFF	256MB	Local to PCI window(s)	–/2	–	Not cachable
0xF002 0000 – 0xF002 FFFF	64KB	PowerSpan internal registers	–/3	–	Not cachable
0xF008 0000 – 0xF008 FFFF	64KB	QuadFALC	–/3	2	Not cachable
0xFF00 0000 – 0xFF01 FFFF	128 KB	MPC internal registers (IMMR initial value)	–/3	–	Not cachable
0xFF80 0000 – 0xFFFF FFFF	8 MB	FLASH (initial vector table at 0xFFF0 0000)	3/3	0	Cachable

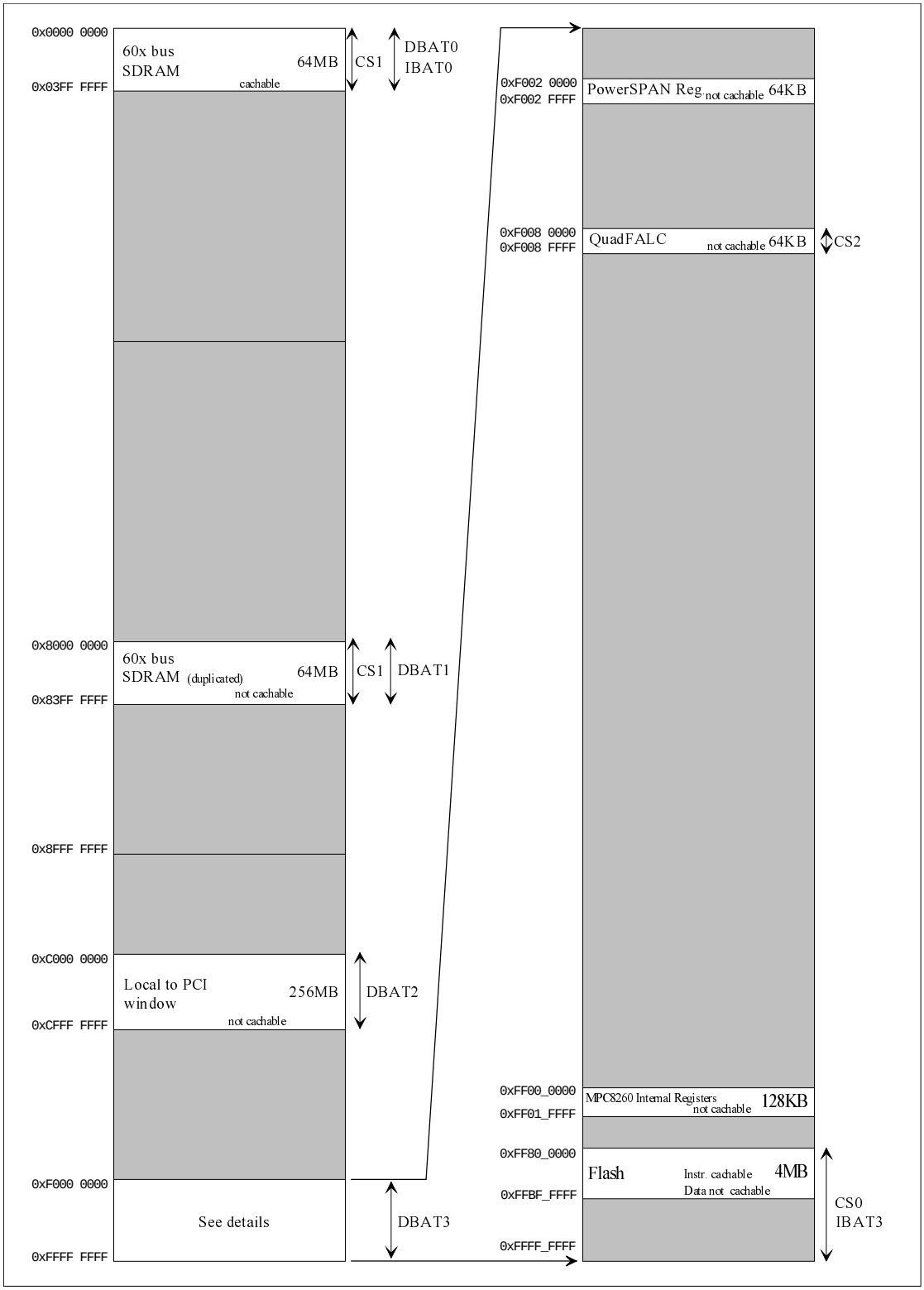


Figure 1-2. Local Space Mapping

NOTE

Accesses from the CPM and the PowerSpan cannot go through the Memory Management Unit (MMU), unlike the core accesses. Therefore, the CPM and PowerSpan must use the physical addresses when accessing the SDRAMs (most significant bit = 0). Accesses to 0x8000 0000 will not address the SDRAMs.

Interrupts

The PCI bridge PowerSpan and the communication peripherals generate interrupt requests to the PowerQUICC II. These interrupts are level sensitive, active low.

Table 1-2. Local Interrupts

Source	MPC2860 Pin	MPC8260 IRQ
PowerSpan interrupt (ATN)	–IRQ1/DP1/–EXT_BG2	–IRQ1
QuadFALC Interrupt	–IRQ2/DP2/–TLBISYNC/–EXT_DBG2	–IRQ2
Ethernet LIU (LXT971A) interrupt	–IRQ3/DP3/–CKSTP_OUT/–EXT_BR3	–IRQ3

Memory Controllers

The sophisticated memory controller units included in the PowerQUICC II are used on the 4538 boards to control all the external devices, except the PowerSpan, which is directly a 60x bus-compatible device. These units are a General Purpose Chip-select Machine (GPCM) for SRAM, FLASH, and peripherals control, three User Programmable Machines (UPM), and two SDRAM control machines (one used for Main SDRAM on 4538).

The memory controller unit to be used is defined bank per bank. Each bank is defined by its Base Register (BRx) and its Option Register (ORx). The memory machine selection is done in the Option register.

Table 1-3. PowerQUICC II Memory Controller Machine Usage

Element Accessed	Bank	Memory Controller
FLASH EEPROM	0	60x bus GPCM
60x bus Main memory	1	60x bus SDRAM machine
QuadFALC	2	60x bus UPMA

Communication Processor Module (CPM) I/O Ports

The CPM part of the PowerQUICC II provides several communication functions. These functions use multi-mode pins that are grouped in four I/O ports: Port A, B, C, and D. The 4538 communications controller uses these ports as shown in the following tables:

Table 1-4. CPM Port A Usage

CPM I/O Port	Pin Configuration	Dir	Usage
PA(0)	Output	O	COMCLK_N signal
PA(1:5)	Output	O	Unused
PA(6)	TDMa1:L1RSYNC	I	TDMa1
PA(7)	Output	O	SWMODE_N signal
PA(8)	TDMa1:L1RXD	I	TDMa1
PA(9)	TDMa1:L1TXD	O	TDMa1
PA(10:31)	Output	O	Unused

Table 1-5. CPM Port B Usage

CPM I/O Port	Pin Configuration	Dir	Usage
PB(4:7)	FCC3:TXD[3:0]	O	Fast Ethernet
PB(8:11)	FCC3:RXD[0:3]	I	Fast Ethernet
PB(12)	FCC3:CRS	I	Fast Ethernet
PB(13)	FCC3:COL	I	Fast Ethernet
PB(14)	FCC3:TX_EN	O	Fast Ethernet
PB(15)	FCC3:TX_ER	O	Fast Ethernet
PB(16)	FCC3:RX_ER	I	Fast Ethernet
PB(17)	FCC3:RX_DV	I	Fast Ethernet
PB(18:19)	Output	O	Unused
PB(20)	TDMd2:L1RSYNC	I	TDMd2
PB(21)	Output	O	Unused
PB(22)	TDMd2:L1RXD	I	TDMd2
PB(23)	TDMd2:L1TXD	O	TDMd2
PB(24)	TDMc2:L1RSYNC	I	TDMc2
PB(25)	Output	O	Unused
PB(26)	TDMc2:L1RXD	I	TDMc2
PB(27)	TDMc2:L1TXD	O	TDMc2
PB(28)	Output	O	Unused
PB(29)	TDMb2:L1RSYNC	I	TDMb2
PB(30)	TDMb2:L1RXD	I	TDMb2
PB(31)	TDMb2:L1TXD	O	TDMb2

Table 1-6. CPM Port C Usage

CPM I/O Port	Pin Configuration	Dir	Usage
PC(0:1)	Output	O	Unused
PC(2)	Output	O	QuadFALC reset (0=active)
PC(3:15)	Output	O	Unused
PC(16)	CLK16	I	Fast Ethernet Tx Clock
PC(17)	CLK15	I	TDMd1,TDMb2 Clock
PC(18)	CLK14	I	Fast Ethernet Rx Clock
PC(19)	CLK13	I	TDMc1,TDMa2 Clock
PC(20)	Output	O	Unused
PC(21)	BRG6	O	12.500 MHz to QuadFALC
PC(22)	Output	O	Unused
PC(23)	Output	O	Unused
PC(24)	Output	O	LXT971 reset (0=active)
PC(25)	Output	O	Fast Ethernet MDC
PC(26)	Input/Output	I/O	Fast Ethernet MDIO
PC(27)	CLK5	I	25.000 MHz Clock Input
PC(28)	Output	O	Unused
PC(29)	CLK3	I	TDMb1,TDMc2 Clock
PC(30)	Output	O	Unused
PC(31)	CLK1	I	TDMa1,TDMd2 Clock

Table 1-7. CPM Port D Usage

CPM I/O Port	Pin Configuration	Dir	Usage
PD(4:7)	Output	O	Unused
PD(8)	SMC1:SMRxD	I	SMC UART
PD(9)	SMC1:SMTxD	O	SMC UART
PD(10)	TDMb1:L1RSYNC	I	TDMb1
PD(11)	Output	O	Unused
PD(12)	TDMb1:L1RXD	I	TDMb1
PD(13)	TDMb1:L1TXD	O	TDMb1
PD(14)	Output	O	CPU_LED2 (User-programmable)
PD(15)	Output	O	CPU_LED1 (User-programmable)
PD(16)	Output	O	CPU_LED6 (User-programmable)
PD(17)	Output	O	CPU_LED4 (User-programmable)
PD(18)	Output	O	CPU_LED3 (User-programmable)
PD(19)	Output	O	Unused
PD(20)	TDMa2:L1RSYNC	I	TDMa2

Table 1-7. CPM Port D Usage (cont)

CPM I/O Port	Pin Configuration	Dir	Usage
PD(21)	TDMa2:L1RXD	I	TDMa2
PD(22)	TDMa2:L1TXD	O	TDMa2
PD(23)	TDMd1:L1RSYNC	I	TDMd1
PD(24)	TDMd1:L1RXD	I	TDMd1
PD(25)	TDMd1:L1TXD	O	TDMd1
PD(26)	TDMc1:L1RSYNC	I	TDMc1
PD(27)	TDMc1:L1RXD	I	TDMc1
PD(28)	TDMc1:L1TXD	O	TDMc1
PD(29:31)	Output	O	Unused

CAUTION

The I/O ports described as "Unused" in the tables above must be configured as general purpose outputs (the logical level does not matter) in order to avoid their electrical level to float.

CPM TDM Busses

The CPM in the MPC8260 features two Serial Interfaces, each one featuring four TDM busses, for a total of eight TDM busses (TDMa1 ... TDMd1, and TDMa2 ... TDMd2). For all the TDM busses used, clock and frame are common for receive and transmit directions (configured in the SIxxMR registers).

Table 1-8. CPM SI1 TDM Busses Wiring

TSA Signal	TDMa1	TDMb1	TDMc1	TDMd1	Dir
L1RTCLK	PC(31)	PC(29)	PC(19)	PC(17)	I
L1RTSYNC	PA(6)	PD(10)	PD(26)	PD(23)	I
L1RXD	PA(8)	PD(12)	PD(27)	PD(24)	I
L1TXD	PA(9)	PD(13)	PD(28)	PD(25)	O

Table 1-9. CPM SI2 TDM Busses Wiring

TSA Signal	TDMa2	TDMb2	TDMc2	TDMd2	Dir
L1RTCLK	PC(19)	PC(17)	PC(29)	PC(31)	I
L1RTSYNC	PD(20)	PB(29)	PB(24)	PB(20)	I
L1RXD	PD(21)	PB(30)	PB(26)	PB(22)	I
L1TXD	PD(22)	PB(31)	PB(27)	PB(23)	O

The two first TDM busses of each serial interface are connected to the four TDM busses of the QuadFALC. The two others TDM busses of each serial interface are used in “pass through mode”. The TDM busses are at a bit rate of 2.048 Mb/s or 8.192 Mb/s.

Bank of Clocks

The PowerQUICC II CPM features a bank of clocks that can be selected independently for each device used. However, the choice for each device is limited. In addition to the ports configuration as clock inputs, it is necessary to configure the clock source of each TDM bus. For all the TDM busses used, the clock is common for receive and transmit directions (configured in the SIxxMR registers).

Table 1-10. CPM Bank of Clocks Usage

Clock	CPM I/O Port	Usage
CLK1	PC(31)	TDMa1, TDMd2
CLK3	PC(29)	TDMb1, TDMc2
CLK5	PC(27)	25.000 MHz for BRG6
CLK13	PC(19)	TDMc1, TDMa2
CLK14	PC(18)	Fast Ethernet Rx Clock
CLK15	PC(17)	TDMd1, TDMb2
CLK16	PC(16)	Fast Ethernet Tx Clock

Baud Rate Generator

The Baud Rate Generator receives CLK5 = 25.000 MHz ±20 ppm and provides BRG6 = 12.500 MHz ±20 ppm for the QuadFALC clock input.

Table 1-11. CPM Baud Rate Usage

Clock	CPM I/O Port	Usage
BRG6	PC(21)	12.5 MHz +/-20ppm to QuadFALC

Ethernet 10/100BaseT

The FCC3 part of the CPM is used to control an Ethernet 10/100baseT port. An on-board LXT971A line interface unit controls the Ethernet interface to a RJ45 connector J3. The CPM interface to the line interface unit is a MII (Media-Independent Interface) bus.

Table 1-12. Ethernet Signals on the CPM

Ethernet Signal	CPM I/O Port	Dir	Description
FE_TXD[3:0]	PB(4:7)	O	Transmit Nibble Data
FE_RXD[0:3]	PB(8:11)	I	Receive Nibble Data
FE_CRS	PB(12)	I	Carrier Sense

Table 1-12. Ethernet Signals on the CPM (cont)

Ethernet Signal	CPM I/O Port	Dir	Description
FE_COL	PB(13)	I	Collision Detect
FE_TX_EN	PB(14)	O	Transmit Enable
FE_TX_ER	PB(15)	O	Transmit Error
FE_RX_ER	PB(16)	I	Receive Error
FE_RX_DV	PB(17)	I	Receive Data Valid
FE_MDC	PC(25)	O	Management Data Clock
FE_MDIO	PC(26)	I/O	Management Data I/O
FE_TX_CLK	PC(16)	I	Transmit clock
FE_RX_CLK	PC(18)	I	Receive Clock
-FE_RESET	PC(24)	O	LXT971 reset control
-FE_INT	IRQ3	I	LXT971 interrupt

Three Ethernet LEDs, LED3, LED4, and LED5, driven respectively by the LXT971A LED/CFG(1:3) outputs, are provided on the front panel.

TTY Console Serial Port

The SMC1 part of the CPM is used as a simple asynchronous serial port for connection to a TTY console. An on-board RS232 transceiver translates the signals to RS232 electrical levels which are routed to the 2.5mm stereo jack connector J4.

Table 1-13. Asynchronous Console Serial Port Wiring

SMC1 Signal	CPM I/O Port	Dir	J4 Connector
GND	–	–	Ring
TXD	PD(9)	O	Tip
RXD	PD(8)	I	Sleeve

User-Programmable LEDs

Five user-programmable LEDs (CPU_LEDs) are provided: four on the board, and one on the front panel. They are controlled through CPM I/O ports used as simple outputs.

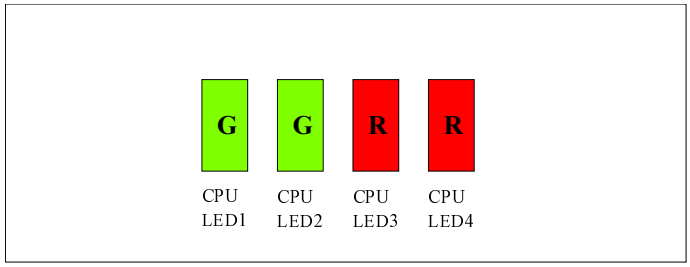


Figure 1-3. Board CPU_LEDs

Table 1-14. User-Programmable LED Control Ports

CPM I/O	Signal Name	Description
PD(14)	CPU_LED2	Board user-programmable green LED (CPU LED2) control. 0=On, 1=Off.
PD(15)	CPU_LED1	Board user-programmable green LED (CPU LED1)control. 0=On, 1=Off.
PD(16)	CPU_LED6	Front panel user-programmable green LED (LED6) control. 1=On, 0=Off.
PD(17)	CPU_LED4	Board user-programmable red LED (CPU LED4) control. 0=On, 1=Off.
PD(18)	CPU_LED3	Board user-programmable red LED (CPU LED3) control. 0=On, 1=Off.

The PCI Bridge

A dedicated PCI bridge, the Tundra PowerSpan, controls the interface between the card and the host 32-bit PCI bus.

The PowerSpan implements all the registers needed by the PCI 2.2 standard, providing the Plug-and-Play capability, as well as the Hot-Swap Friendly capabilities. It supports Target and Master accesses between the PCI bus and the local 60x bus.

It also implements windows and different mechanisms to interface between the PCI host and the card. Exchanges can use the following elements from the PowerSpan:

- Runtime registers (mailboxes, doorbells, semaphores)
- Four memory windows from the PCI memory space to the Local memory space
- Eight memory windows from the Local space to the PCI memory or I/O space
- Four independent bidirectional DMA engines
- An I²O messaging unit

This chip implements FIFO buffers for all the exchanges through the different windows between the two buses, so that the local bus clock is independent from the PCI bus clock.

All the PowerSpan internal registers are grouped in a 4 KB memory space that can be accessed by the PCI host and the local processor. On the PCI side, the PCI base address of this register space is defined by PCI configuration register PCIBAR1 (offset 0x14). On the local side, the local base address has been conventionally fixed to 0xF0020000.

The PowerSpan internal register set can be split into six different functional groups:

- PCI configuration registers (these registers, defined by the PCI specification, can be accessed in the standard PCI configuration space or in the local PowerSpan internal registers space)
- PCI registers
- Processor bus registers
- DMA registers
- Miscellaneous registers (Mailboxes, Doorbells, Interrupts, Semaphores)
- I²O messaging registers

The details of the PowerSpan registers can be found by consulting the PowerSpan data sheet available in the Tundra Web site.

PowerSpan PCI Configuration Registers

As defined by the PCI specification, the communications controller has a unique 256-byte memory space, called configuration space, that maps all the PCI configuration registers. Access to this area is done through CompactPCI Configuration Read and PCI Configuration Write cycles.

Table 1-15. PCI Configuration Registers

Register	Size	PCI cfg Address	Local Offset	Description
PCIIDR	32	0x00	0x00	Vendor and Device Identification
PCICR	16	0x04	0x06	PCI Command
PCISR	16	0x06	0x04	PCI Status
PCIREV	8	0x08	0x0B	Revision Identification
PCICCR	24	0x09	0x08	Class Code
PCICLSR	8	0x0C	0x0F	Cache Line Size
PCILTR	8	0x0D	0x0E	Master Latency Timer
PCIHTR	8	0x0E	0x0D	Header Type
PCIBISTR	8	0x0F	0x0C	Built-in Self Test
PCIBAR0	32	0x10	0x10	I²O registers base address
PCIBAR1	32	0x14	0x14	PowerSpan internal registers base address
PCIBAR2	32	0x18	0x18	PCI-to-Local Window 0 PCI base address
PCIBAR3	32	0x1C	0x1C	PCI-to-Local Window 1 PCI base address
PCIBAR4	32	0x20	0x20	PCI-to-Local Window 2 PCI base address
PCIBAR5	32	0x24	0x24	PCI-to-Local Window 3 PCI base address
PCISVID	16	0x2C	0x2E	Subsystem Vendor ID

Table 1-15. PCI Configuration Registers (cont)

Register	Size	PCI cfg Address	Local Offset	Description
PCISID	16	0x2E	0x2C	Subsystem Device ID
PCICAP	8	0x34	0x37	Capabilities pointer
PCIILR	8	0x3C	0x3F	Interrupt Line
PCIIPR	8	0x3D	0x3E	Interrupt Pin
PCIMGR	8	0x3E	0x3D	Minimum Grant
PCIMLR	8	0x3F	0x3C	Maximum Latency
HS_CSR	32	0xE4	0xE4	Hot Swap Control and Status Register
VPD_CSR	32	0xE8	0xE8	PCI Vital Product Control/Status Register
VPD_D	32	0xEC	0xEC	PCI Vital Product Data-Data Register

These registers are initialized with fixed reset values or with values stored in the I²C serial EEPROM and then used by the PCI HOST, mainly during the Power-On Self Test (POST) for plug and play functionality or later by the operating system for enumeration.

The PCI Vendor ID equals 0x107E, and the PCI Device ID equals 0x9070.

PowerSpan PCI Registers

These registers are used to define the parameters of the PCI-to-Local windows. They are mapped in the PCI memory space (base address defined in PCI configuration register 0x14 PCIBAR1) and in the local space for the local processor (base address 0xF0020000).

Table 1-16. PowerSpan PCI Registers

Offset	Register	Description
0x100	P1_TI0_CTL	PCI Target Image 0 Control Register
0x104	P1_TI0_TADDR	PCI Target Image 0 Translation Address Register
0x110	P1_TI1_CTL	PCI Target Image 1 Control Register
0x114	P1_TI1_TADDR	PCI Target Image 1 Translation Address Register
0x120	P1_TI2_CTL	PCI Target Image 2 Control Register
0x124	P1_TI2_TADDR	PCI Target Image 2 Translation Address Register
0x130	P1_TI3_CTL	PCI Target Image 3 Control Register
0x134	P1_TI3_TADDR	PCI Target Image 3 Translation Address Register
0x150	P1_ERRCS	PCI Bus error control and status register
0x154	P1_AERR	PCI Address error log register
0x160	P1_MISC_CSR	PCI Miscellaneous Control and Status Register

Table 1-16. PowerSpan PCI Registers (cont)

Offset	Register	Description
0x164	P1_ARB_CTRL	PCI Bus Arbiter Control Register

PowerSpan Processor Bus Registers

These registers are used to define the parameters of the local to PCI windows. They are mapped in the PCI memory space (base address defined in PCI configuration register 0x14 PCIBAR1) and in the local space for the local processor (base address 0xF0020000).

Table 1-17. PowerSpan Processor Bus Registers

Offset	Register	Description
0x200	PB_SI0_CTL	Processor Bus Slave Image 0 Control Register
0x204	PB_SI0_TADDR	Processor Bus Slave Image 0 Translation Address Register
0x208	PB_SI0_BADDR	Processor Bus Slave Image 0 Base Address Register
0x210	PB_SI1_CTL	Processor Bus Slave Image 1 Control Register
0x214	PB_SI1_TADDR	Processor Bus Slave Image 1 Translation Address Register
0x218	PB_SI1_BADDR	Processor Bus Slave Image 1 Base Address Register
0x220	PB_SI2_CTL	Processor Bus Slave Image 2 Control Register
0x224	PB_SI2_TADDR	Processor Bus Slave Image 2 Translation Address Register
0x228	PB_SI2_BADDR	Processor Bus Slave Image 2 Base Address Register
0x230	PB_SI3_CTL	Processor Bus Slave Image 3 Control Register
0x234	PB_SI3_TADDR	Processor Bus Slave Image 3 Translation Address Register
0x238	PB_SI3_BADDR	Processor Bus Slave Image 3 Base Address Register
...	...	...
0x270	PB_SI7_CTL	Processor Bus Slave Image 7 Control Register
0x274	PB_SI7_TADDR	Processor Bus Slave Image 7 Translation Address Register
0x278	PB_SI7_BADDR	Processor Bus Slave Image 7 Base Address Register
0x280	PB_REG_BADDR	Processor Bus Register Image Base Address Register
0x290	PB_CONF_INFO	Processor Bus PCI Configuration Cycle Information Register
0x294	PB_CONF_DATA	Processor Bus PCI Configuration Cycle Data Register
0x2A0	PB_P1_IACK	Processor Bus to PCI Interrupt Acknowledge Cycle Register
0x2B0	PB_ERRCS	Processor Bus Error Control and Status Register
0x2B4	PB_AERR	Processor Bus Address Error Log Register
0x2C0	PB_MISC_CSR	Processor Bus Miscellaneous Control and Status Register
0x2D0	PB_ARB_CTRL	Processor Bus Arbiter Control Register

PowerSpan DMA Registers

These registers are used to control the four bidirectional DMA engines provided in the PowerSpan. They are mapped in the PCI memory space (base address defined in PCI configuration register 0x14 PCIBAR1) and in the local space for the local processor (base address 0xF0020000).

Table 1-18. PowerSpan DMA Registers

Offset	Register	Description
0x304	DMA0_SRC_ADDR	DMA0 Source Address Register
0x30C	DMA0_DST_ADDR	DMA0 Destination Address Register
0x314	DMA0_TCR	DMA0 Transfer Control Register
0x31C	DMA0_CPP	DMA0 Command Packet Pointer Register
0x320	DMA0_ GCSR	DMA0 General Control Register
0x324	DMA0_ATTR	DMA0 Attributes Register
0x334	DMA1_SRC_ADDR	DMA1 Source Address Register
0x33C	DMA1_DST_ADDR	DMA1 Destination Address Register
0x344	DMA1_TCR	DMA1 Transfer Control Register
0x34C	DMA1_CPP	DMA1 Command Packet Pointer Register
0x350	DMA1_ GCSR	DMA1 General Control Register
0x354	DMA1_ATTR	DMA1 Attributes Register
0x364	DMA2_SRC_ADDR	DMA2 Source Address Register
0x36C	DMA2_DST_ADDR	DMA2 Destination Address Register
0x374	DMA2_TCR	DMA2 Transfer Control Register
0x37C	DMA2_CPP	DMA2 Command Packet Pointer Register
0x380	DMA2_ GCSR	DMA2 General Control Register
0x384	DMA2_ATTR	DMA2 Attributes Register
0x394	DMA3_SRC_ADDR	DMA3 Source Address Register
0x39C	DMA3_DST_ADDR	DMA3 Destination Address Register
0x3A4	DMA3_TCR	DMA3 Transfer Control Register
0x3AC	DMA3_CPP	DMA3 Command Packet Pointer Register
0x3B0	DMA3_ GCSR	DMA3 General Control Register
0x3B4	DMA3_ATTR	DMA3 Attributes Register

PowerSpan Miscellaneous Registers

This group of registers includes several configuration registers for the interrupt functions, as well as various runtime registers: mailboxes, doorbells, interrupt control/status, and semaphores. They are mapped in the PCI memory space (base address defined in PCI configuration register 0x14 PCIBAR1) and in the local space for the local processor (base address 0xF0020000).

Table 1-19. PowerSpan Miscellaneous Registers

Offset	Register	Description
0x400	MISC_CSR	Miscellaneous Control/Status Register
0x404	CLOCK_CTL	Clock Control Register
0x408	I²C_CSR	I²C Interface Control and Status Register
0x40C	RST_CSR	Reset Control and Status Register
0x410	ISR0	Interrupt Status Register 0
0x414	ISR1	Interrupt Status Register 1
0x418	IER0	Interrupt Enable Register 0
0x41C	IER1	Interrupt Enable Register 1
0x420	IMR_MBOX	Interrupt Map Register: Mailbox
0x424	IMR_Db	Interrupt Map Register: Doorbell
0x428	IMR_DMA	Interrupt Map Register: DMA
0x42C	IMR_HW	Interrupt Map Register: Hardware
0x430	IMR_P1	Interrupt Map Register: PCI
0x438	IMR_PB	Interrupt Map Register: Processor Bus
0x43C	IMR_PB2	Interrupt Map Register 2: Processor Bus
0x440	IMR_MISC	Interrupt Map Register: Miscellaneous
0x444	IDR	Interrupt Direction Register
0x450–0x46C	MBOX0 – MBOX7	Mailbox 0 to 7 Registers
0x470	SEMA0	Semaphore 0 Register
0x474	SEMA1	Semaphore 1 Register

PowerSpan I²O Registers

The PowerSpan includes I²O messaging queues controlled by several registers. These registers are mapped in two places in the PCI memory space: at the base address defined in the PCI configuration register 0x10 PCIBAR0 and in the PowerSpan internal register space (base address defined in PCI configuration register 0x14 PCIBAR1). They are also mapped in the local space for the local processor (base address 0xF0020000).

Table 1-20. PowerSpan I²O Registers

Offset	Register	Description
0x500	PCI_TI2O_CTL	PCI I²O Target Image Control Register
0x504	PCI_TI2O_TADDR	PCI I²O Target Image Translation Address Register
0x508	I2O_CSR	I2O Control and Status Register
0x50C	I2O_QUEUE_BS	I²O Queue Base Address
0x510	IFL_BOT	I²O Inbound Free List Bottom Pointer Register
0x514	IFL_TOP	I²O Inbound Free List Top Pointer Register
0x518	IFL_TOP_INC	I²O Inbound Free List Top Pointer Increment Register
0x51C	IPL_BOT	I²O Inbound Post List Bottom Pointer Register
0x520	IPL_BOT_INC	I²O Inbound Post List Bottom Pointer Increment Register
0x524	IPL_TOP	I²O Inbound Post List Top Pointer Register
0x528	OFL_BOT	I²O Outbound Free List Bottom Pointer Register
0x52C	OFL_BOT_INC	I²O Inbound Free List Bottom Pointer Increment Register
0x530	OFL_TOP	I²O Outbound Free List Top Pointer Register
0x534	OPL_BOT	I²O Outbound Post List Bottom Pointer Register
0x538	OPL_TOP	I²O Outbound Post List Top Pointer Register
0x53C	OPL_TOP_INC	I²O Outbound Post List Top Pointer Increment Register
0x540	HOST_OIO	I²O Host Outbound Index Offset Register
0x544	HOST_OIA	I²O Host Outbound Index Alias Register
0x548	IOP_OI	I²O IOP Outbound Index Register
0x54C	IOP_OI_INC	I²O IOP Outbound Index Increment Register

Interrupt Pins and Doorbell Usage

The PowerSpan provides one interrupt pin on the PCI side (–INTA) and six other interrupt pins (–INT0 to –INT5) on the local side. On the 4538, only –INTA and –INT0 are used for true interrupt functions. The five other pins are used as I/O pins to control several signals.

The PowerSpan offers the ability to map any interrupt source to any interrupt pin. This capability is used to divert interrupts –INT1 to –INT5 from a pure interrupt function usage.

Interrupt pins –INT1 to –INT4 are configured as output ports and conventionally associated with doorbell bits DB3 to DB6 in the PowerSpan. Each doorbell bit, when set, will activate its corresponding interrupt pin (level = 0), and when reset will deactivate it (level =1).

Interrupt pin –INT5 is used as an input. Its state can be read in the PowerSpan Interrupt status register. As an interrupt source, it was decided not to map it to any interrupt output, so it will not generate interrupts. As an interrupt output pin, it was decided not to associate it to any interrupt source.

Interrupt pins –INTA and –INT0, used for true interrupt functions, have several other interrupt sources, such as Mailboxes interrupts, DMA interrupts, I²O interrupts, PCI bus or local bus error interrupts, etc. They are conventionally associated with a doorbell bit for software activation capability.

Table 1-21. PowerSpan Interrupt Pin Usage

PowerSpan Pin	Doorbell	Dir	Signal Name	Usage
–INTA	DB0	O	–INTA	Interrupt from the 4538 to the PCI Host controlled by software by the PowerQUICC II.
–INT0	DB2	O	–INTPSP	Interrupt from the powerSpan to the PowerQUICC II interrupt input –IRQ1
–INT1	DB3	O	–INTRST	Flash mode: When this output is set to 0, the PowerQUICC II is put in reset, its busses are tri-stated, the 60x bus is parked for the PowerSpan and the special address translation mode on the FLASH memory is enabled.
–INT2	DB4	O	–PSP_INT2/ –HRESET/ ISPDI	When this pin is set to 0, the PowerQUICC II hardware reset signal is activated. For ISP EPLD programming, this pin serves also as "Serial Data In" signal.
–INT3	DB5	O	–PSP_INT3/ –SRESET/ ISPMODE	When this pin is set to 0, the PowerQUICC II software reset signal is activated. For ISP EPLD programming, this pin serves also as "Mode" signal.
–INT4	DB6	O	–PSP_INT4/ ISPCK	For ISP EPLD programming, this pin serves as "Serial Clock" signal.
–INT5	–	I	–PSP_INT5/ ISPDO	For ISP EPLD programming, this pin serves as "Serial Data Out" signal.

PCI to Local Interrupt (ATN)

The PowerSpan Interrupt pin –INT0 is used to control the PCI-to-Local Interrupt (renamed ATN (Attention) in the software examples).

Local to PCI Interrupt (–INTA)

The PowerQUICC II can generate an interrupt toward the PCI Host by setting a doorbell bit. Conventionally, doorbell bit 0 has been dedicated to this task, and has been associated with the PCI interrupt pin –INTA in the PowerSpan Interrupt Map registers.

Hardware and Software Resets Through the PowerSpan

PowerSpan interrupt pins –INT2 and –INT3 are used as output ports to control the MPC8260 hardware reset signal –HRESET and software reset signal –SRESET respectively. The PowerSpan Interrupt Map registers must have previously been correctly initialized.

During a power-up sequence, –HRESET and –SRESET are first activated and then deactivated once the PCI bus reset signal is deactivated. This allows the PowerQUICC II to boot without any host intervention, just after the end of the PCI reset.

For a normal utilization, the card should be reset by the PCI host (if needed) using only the –SRESET signal. The –HRESET signal is used for special cases, such as FLASH memory reprogramming through PCI.

Local Space Access From PCI Memory Space

The PowerSpan provides four memory windows from the PCI memory space to the Local memory space. Each window can map a programmable size of the local memory space into the PCI memory space. The size of the windows and their enabling is set in the PowerSpan registers P1_TIx_CTL, and preset at power-up by the serial EEPROM.

In the 4538 communications controller, only two windows are enabled. They have been set to a relatively small size (2 MB and 512 KB), in order to comply with high availability operating system requirements. These operating systems are able to do dynamic PCI re-configuration during hot swap, only if the total memory size requested by the board is not too big.

The PCI base address of each window is defined in a PCI configuration register. Window 0 base address is set in P1_BAR2, Window 1 base address is set in P1_BAR3, etc. Each window can be moved on the local memory space, using a PowerSpan translation register (P1_TIx_TADDR), so that even a small window can allow access to any part of the 4 GB of local memory space.

During a PCI host access to the local space, the high-order address bits of the local bus must be generated by the PowerSpan (as defined in the PowerSpan P1_TI0_TADDR register) and the low-order address bits of the local bus come from the PCI address. This mode is called “Address Translation” in the PowerSpan Manual.

NOTE

A PowerSpan PCI-to-Local window must have been enabled in the I²C serial EEPROM, in order to allow the CompactPCI host to detect it at system power-on or after the "Hot Swap insertion" of the board and to map it in the PCI space. The corresponding PowerSpan register "PCI Target Image Control Register" must also have been initialized with the "Image Enable" bit set (IMG_EN=1) and the address translation mechanism enabled (TA_EN=1).

Figure 1-4 on page 23 illustrates the PCI-to-Local window mechanism.

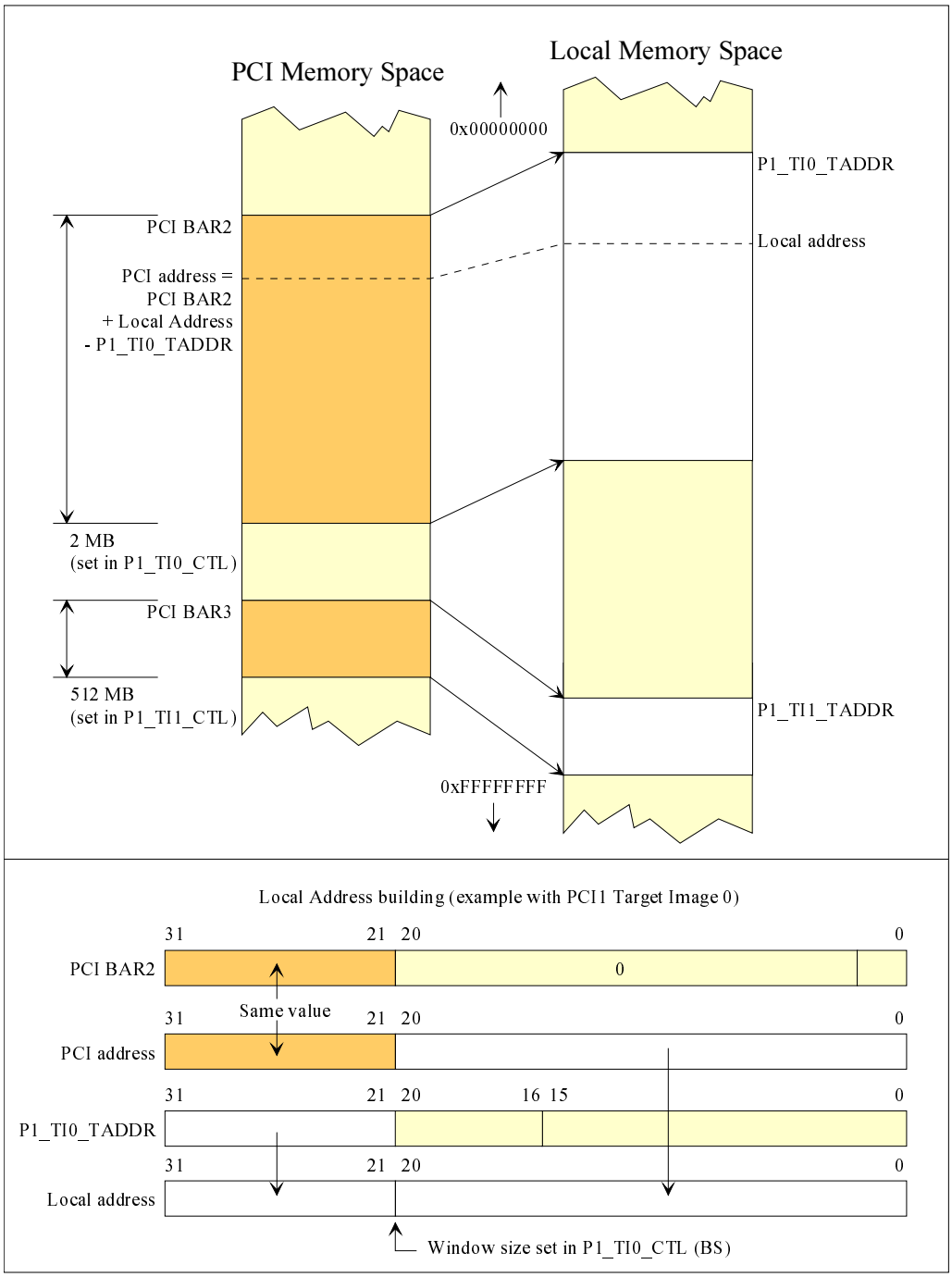


Figure 1-4. Local Space Access From PCI Memory Space

When the processor is running, the PCI bus can access all the elements connected to the local bus, except the FLASH boot memory. The accessible elements are the main SDRAM memory (the processor's SDRAM memory controller must be initialized), the processor

dual port RAM, the QuadFALC framers, and the IMA device. (the processor must have its chip selects programmed). The local space mapping is the same as when accessed by the processor (see *PCI Local Space Mapping* on page 5).

It is not possible to have access to the entire FLASH device when the processor is running, because the FLASH device is an 8-bit data bus device connected to the 64-bit-only local bus of the PowerSpan. Only bytes modulo 8 are reachable.

This problem has been neutralized for the other non-64-bit peripherals, by tying their peripheral address bits 0 to N to local address bits 3 to N+3 respectively, so that all their registers can be accessed on byte lane 0, at consecutive modulo 8 addresses.

When the processor is in the reset state, its memory controllers and chip-select signals are reset, so nothing can be accessed, except the FLASH memory, for which a special mechanism has been implemented.

NOTE

It is possible to write from the PCI bus through a PowerSpan memory window to the MPC8260 internal registers but it is not possible to read them. When the PowerSpan performs a read on the 60x processor bus, it always generates a full 64-bit read. Because most of the MPC8260 internal registers only respond to byte or word read cycles, the returned value is 0xFFFFFFFF.

Access to the FLASH EEPROM Through CompactPCI

For FLASH in-situ re-programming through CompactPCI, there is a special FLASH mode. In this mode, the PowerQUICC II is reset and logic generates a FLASH chip-select and works around the problem of an 8-bit device connected to a 64-bit-only PowerSpan.

The specific FLASH mode is enabled by one of the PowerSpan interrupt pins (–INT1) used as an output port. When –INT1 is set to 0, the PowerQUICC II is maintained in Hard Reset state (–HRESET=0), its pins are tri-stated, the 60x bus is parked on the PowerSpan, and the following address bus remap is implemented: the FLASH device's low order address bits A(2:0) are driven by the PowerSpan address bits A(24:22). This remap allows full access to the FLASH content through byte lane 0 of the 64-bit 60x bus, provided that some address translation is done by the software.

For more information on FLASH EEPROM device, see *The FLASH EEPROM Boot Memory* on page 29.

PCI Memory Space and I/O Space Access From the PowerQUICC II

The PowerSpan provides eight memory windows from the Local Memory space to the PCI memory space or PCI I/O space. Each window can map a programmable size of the PCI memory or I/O space into the PCI memory space. The size of the windows and their enabling is set in PowerSpan registers PB_SIx_CTL, and preset at power-up: the first window is preset by the serial EEPROM and the seven others are preset as disabled.

On the 4538 board, the serial EEPROM content disables the windows. By default, no Local to PCI window is enabled. It is not recommended using these windows for transfers from or to the PCI local space, because this mechanism can result in bad performance, depending on the other PCI devices tied to the PCI bus.

The local base address of each window is defined in PowerSpan internal register PB_SIx_BADDR. Note that the window must be mapped in the local space between 0xC0000 0000 and 0xCFFF FFFF, in order to comply with the card local space usage. Each window can be moved on the PCI memory or I/O space, using a PowerSpan translation register (PB_SIx_TADDR), so that even a small window can allow access to any part of the PCI space.

During a PowerQUICC II access to the PCI space, the high-order address bits on the PCI bus are generated by the PowerSpan (as defined in the PowerSpan PB_SIx_BADDR register) and the low-order address bits on the PCI bus come from the local address. This mode is called “Address Translation” in the PowerSpan Manual.

NOTE

A PowerSpan Local-to-PCI window must be enabled in the PB_SIx_CTL register. Bits IMG_EN (“Image Enable”) and TA_EN (“address translation enable”) must be set.

Figure 1-5 on page 26 illustrates the Local-to-PCI window mechanism:

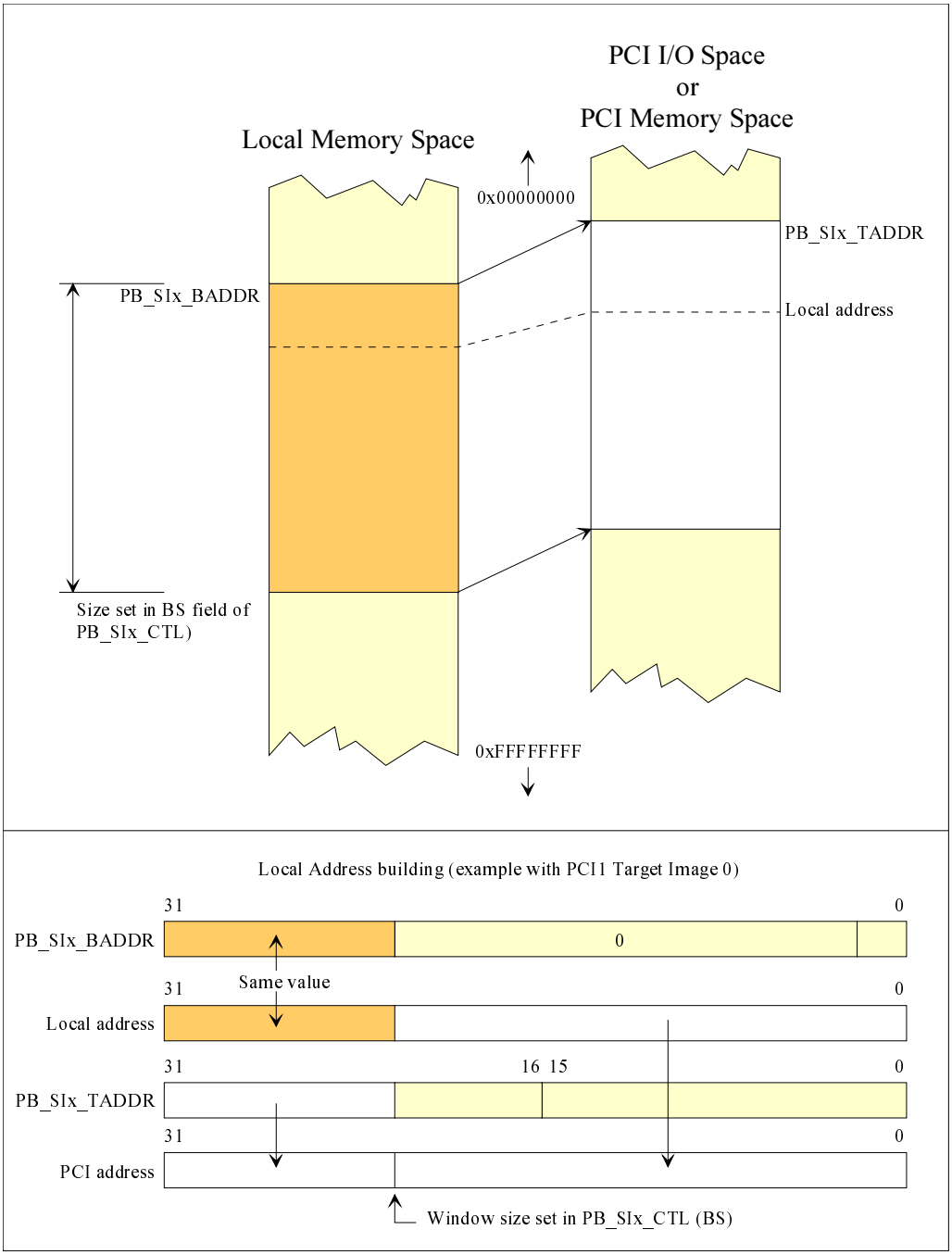


Figure 1-5. PCI I/O or Memory Space Access from Local Space

In-situ EPLDs Programming

Some glue logic is implemented in some EPLDs that can be programmed in-situ through the PCI interface.

These devices keep their programming during power off. So the EPLD should normally be already programmed and the normal user should not be aware of its programming.

The EPLDs are in a daisy-chain configuration, which enables all of them to be programmed at once. They can be programmed in-situ by the PCI host, using some PowerSpan interrupts as I/O pins. A jumper must be placed on board location JP1 to enable the programming (when present, this jumper sets the ISP signal –ISPEN to its active state 0).

Serial EEPROM Connected to the PowerSpan

An I²C serial EEPROM is connected to the PowerSpan. It is used to store some PowerSpan register initialization values and the PCI Vital Product Data (VPD). Other Interphase-specific data is stored there, and there is still some room for other custom data. Table 1-22 shows the memory mapping for the EEPROM.

Table 1-22. Serial EEPROM Mapping

Address	Size	Description
0x00 – 0x3F	64 bytes	PowerSpan registers initial load
0x40 – 0x43	4 bytes	Board Equipment Register
0x44 – 0x8F	76 bytes	VPD and/or Custom data
0x90 – 0xAF	32 bytes	Interphase-Specific Production Data
0xB0 – 0xFF	80 bytes	Boot Monitor parameters

Details about the PowerSpan register initial load are described on Table 2-1 on page 60. Additional information concerning Interphase-specific Production Data and Boot Monitor parameters are provided in the *4538 Built-In Self Test and Monitor Manual*. The VPD and/or Custom Data is available space, for later use)

Board Equipment Register

The “Board Equipment Register” is a 32-bit word that allows the software to precisely determine the board equipment. The first three bytes are common to several Interphase Boards, so many field values are not possible on the 4538. For instance the 4538 does not have Monarch capability, so the Monarch bit will always be set to 0.

Table 1-23. Board Equipment Register Layout

EEPROM Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x40	MPC_ID				FLASH_SIZE		LSDRAM_SIZE	
0x41	SDRAM_SIZE			CAM_SIZE			0	MONARCH
0x42	BUS_FREQ			0	0	0	0	0
0x43	0	0	0	0	0	0	0	ACCESS

Table 1-24. Hardware Configuration Register Field Descriptions

Field	Description
MPC_ID	Microprocessor identifier: 0000: MPC8260ZU200, 200/133/66MHz, rev A.1 0001: MPC8260ZU133, 133/133/66MHz, rev A.1 0010: MPC8260ZU200, 200/133/66MHz, rev B.3 0011-1111: Reserved for future processor versions
FLASH_SIZE	Flash EEPROM size: 00: 1 Mbytes 01: 4 Mbytes 10: 8 Mbytes 11: Reserved for future use
LSDRAM_SIZE	Local SDRAM size: 00: No memory device 01: 8 Mbytes 10: 16 Mbytes 11: Reserved for future use
SDRAM_SIZE	Main SDRAM size: 000: 16 Mbytes (not possible on the 4539) 001: 32 Mbytes 010: 64 Mbytes 011: 128 Mbytes 100-111: Reserved for future use
CAM_SIZE	CAM size: 000: No CAM device 001: 4 K x 64 010: 8 K x 64 (not possible on the 4539) 011: 16 K x 64 100: 32 K x 64 (not possible on the 4539) 101-111: Reserved for future use
MONARCH	Monarch capability: 0: Not monarch capable 1: Monarch capable (not possible on the 4539)
BUS_FREQ	Local Bus frequency: 000: 50.000 MHz 001: 65.536 MHz (the only frequency currently available on the 4539) 010: 66.000 MHz 011-111: Reserved for future use
ACCESS	Access type: 0: 2 port front access board 1: 4 port rear access board

Vital Product Data (VPD)

No VPD has been defined for the 4538.

Interphase-Specific Production Data and Boot Monitor Parameters

Additional information concerning Interphase-specific Production Data and Boot Monitor parameters are provided in the *4538 Built-In Self Test and Monitor Manual*.

The FLASH EEPROM Boot Memory

The boot memory is a 4Mx8 AMD 29LV033 FLASH EEPROM device, placed in the 60x bus byte lane 0. This non-volatile memory device contains the Reset Configuration Word required by the PowerQUICC II during the power-up phase, the 4538 Interphase Boot Firmware Code, and optionally, your own complete operational code. The FLASH memory is always mapped at address 0xFF800000.

Depending on the FLASH memory size, the mapping of the boot firmware will be different. There are three requirements:

- The Reset configuration must be mapped at the beginning of the FLASH memory.
- The initial vector table must be mapped at address 0xFFF00000. This address is never in the FLASH memory, but it will wrap onto its last MB.
- The FLASH Memory is organized in sectors. The reset configuration word and the vector table must be preserved; therefore their entire sectors will be reserved.

The various elements are/must then be mapped as follows (the FLASH addresses are obtained by masking the local address with the flash size: for a 4MB flash device it is 0x003FFFFF).

The flash is mapped from 0xFF800000 to 0xFFFFFFFF (8MByte space). For a 4MB device, the second 4MB space will cover the first one (see below : first and second map).

Table 1-25. FLASH EEPROM Mapping

FLASH Addr	1st MAP	2nd MAP	Size	Description
0x000000	0xFF800000	0xFFC00000	0x00000100	The first 64-kbyte sector of the FLASH contains the Hardware Configuration word (at addresses 0xFF800000, 0xFF800008, 0xFF800010, 0xFF800018). Remaining space is unused.
0x000100	0xFF800100	0xFFC00100	0x0000FE00	Unused byte space.
0x010000	0xFF810000	0xFFC10000	0x002F0000	Free space for Operational Firmware (47*64 KB)
0x300000	0xFFB00000	0xFFF00000	0x00000100	Unused byte space.
0x300100	0xFFB00100	0xFFF00100	0x000358E0*	4538 Boot Firmware ROM code
0x3359E0*	0xFFB359E0*	0xFFF359E0*		Unused byte space.

Table 1-25. FLASH EEPROM Mapping (cont)

FLASH Addr	1st MAP	2nd MAP	Size	Description

* Values depend on Boot Firmware size which vary from one version to another – use the MONITOR INFO command to display the actual size)

The FLASH device is normally controlled by the PowerQUICC II memory controller unit using chip-select signal CS0. The PowerQUICC II can read and re-program the FLASH using the AMD algorithms.

The FLASH device is not intended to be accessed through the CompactPCI bus. Because the FLASH device has an 8-bit data bus, and the PowerSpan supports only 64-bit wide devices, its byte lane can only be accessed by the CompactPCI host for addresses that are multiples of 8.

For more information, see *Access to the FLASH EEPROM Through CompactPCI* on page 24.

The QuadFALC T1/E1/J1 Framer

The 4538 Communication Controller includes one QuadFALC device which controls four independent T1/E1/J1 interfaces. For each interface, the QuadFALC includes a framer and an LIU with data and clock recovery, a frame aligner with two frame elastic buffers for receive clock wander and jitter compensation, a signaling controller with a HDLC controller and 64 bytes deep FIFOs, and an 8-bit micro-processor interface.

Each line can be independently configured for E1 or T1. The pulse shape for CEPT E1 applications is programmed according to ITU-T G.703:

- Data Coding: HDB3
- Voltage of nominal pulse: 3 V (CCITT G703)
- Return Loss Transmitter: –12 dB (CCITT G703)
- Line Impedance: 120 Ohm

The pulse shape for T1 applications is programmed according to ANSI T1.403:

- Data Coding: B8ZS
- Voltage of nominal pulse: 3 V
- Return Loss Transmitter: –3.5 dB
- Line Impedance: 100 Ohm

The pulse shape for J1 applications is programmed according to ITU-T JT G.703:

- Data Coding: B8ZS
- Voltage of nominal pulse: 3 V (TBV)
- Return Loss Transmitter: –3.5 dB (TBV)
- Line Impedance: 110 Ohm

The QuadFALC includes a flexible clock unit that uses a clock supplied on its MCLK pin.

The QuadFALC MCLK input is connected to a 12.500 MHz +/-20ppm fixed frequency (CPM BRG6) used by the internal DPLL. As a result, the GCM registers must be programmed with the following values:

Table 1-26. GCM Register Programming (MCLK=12.500 MHz)

Register	Value
GCM1	0x2B
GCM2	0x5D
GCM3	0xAC
GCM4	0x89
GCM5	0x07
GCM6	0x15

The QuadFALC has an integrated short-haul and long-haul line interface, comprising a receive equalization network, noise filtering, and programmable Line Build-Outs (LBOs). It implements an integrated Channel Service Unit (CSU) in T1 mode. For each type of LBO, the shape of the transmit pulse must be adjusted through its registers LIM0, LIM2, XPM0, XPM1, and XPM2 in order to comply with FCC 68 or ANSI T1.403. Table 1-27 provides the values in T1 mode for the 4538 hardware (in E1 mode, default values are suitable)

Table 1-27. Transmit Pulse Shape Programming

			2 Front Access			4 Rear Access		
Line Build-Out	**LIM0:EQON**	**LIM2:LBO2–1**	**XPM0**	**XPM1**	**XPM2**	**XPM0**	**XPM1**	**XPM2**
E1 Short Haul	0	–	0x39	0x03	0x00	0x9C	0x03	0x00
E1 Long Haul	1	–	0x39	0x03	0x00	0x9C	0x03	0x00
T1 Short Haul (no CSU)	0	00	0x7D	0xA3	0x01	0xBF	0xA3	0x01
T1 Long Haul, 0 dB	1	00	0x7D	0xA3	0x01	0xBF	0xA3	0x01
T1 Long Haul, –7.5 dB	1	01	0xF6	0x02	0x00	0xF6	0x02	0x00
T1 Long Haul, –15 dB	1	10	0xB6	0x01	0x00	0xB6	0x01	0x00
T1 Long Haul –22.5 dB	1	11	0xB4	0x01	0x00	0xB4	0x01	0x00

For each line x, the QuadFALC provides four transmit multifunction ports (XPA_x, XPB_x, XPC_x and XPD_x) and four receive multifunction ports (RPA_x, RPB_x, RPC_x and RPD_x). The tables below indicate how they are used on the 4538 (The RPD port is detailed for each port, since its use differs from one port to another).

Table 1-28. QuadFALC Multifunction Port Usage

QuadFALC port	Dir	Function	Usage
XPA_x	Input	SYPX	TDM bus Frame synchronization pulse
XPB_x	-	-	(Unused)
XPC_x	-	-	(Unused)
XPD_x	-	-	(Unused)
RPA_x	Input/Output	SYPR/RFM	TDM bus Frame synchronization pulse (the function used depends on the TDM bus configuration. See *TDM Bus Configurations* on page 34)
RPB_x	-	-	(Unused)
RPC_x	-	-	(Unused)
RPD_1	Output	$\overline{\text{RFSP}}$ or RMFB	LED1 control (0=On, 1=Off) See Note
RPD_2	Output	$\overline{\text{RFSP}}$ or RMFB	LED2 control (0=On, 1=Off) See Note
RPD_3	-	-	(unused)
RPD_4	-	-	(unused)

NOTE

The two synchronization green LEDs on the front panel are controlled by the RPD_1 and RPD_2 pins configured as $\overline{\text{RFSP}}$ respectively. These LEDs can also be controlled by software, by configuring RPD_1 and RPD_2 pins as RMFB and forcing them to 0 or 1.

The local processor and the PCI host see the QuadFALC as an 8-bit peripheral including a set of 1024 directly addressable registers. These registers are placed at contiguous modulo 8 addresses, starting at addresses 0xF008 0000. The QuadFALC controls its own interrupt line to the local processor.

The QuadFALC reset input is controlled by a PowerQUICC II CPM I/O port PC(2), (0=reset active).

The QuadFALC controls its own interrupt line to the local processor.

Each line of the QuadFALC framers can be configured independently in Line Termination mode (LT) or in Network Termination mode (NT). In the LT mode, the QuadFALC is in slave mode and synchronizes on the lines. In the NT mode, the QuadFALC is in master mode and synchronizes on a reference signal provided through connector P4 or on a free running internal frequency.

On the front access board, the framers 1 and 2 are tied respectively to J1 and J2 connectors. On the rear access board, the framers 1, 2, 3 and 4 are respectively tied to the lines 0, 1, 2 and 3 on P4 connector.

Additional details about the Infineon PEB22554 can be found at Infineon's web site.

The Ethernet Transceiver

The Intel LXT971A is an IEEE compliant Fast Ethernet transceiver for 100-Base-TX and 10-Base-T applications. It is connected to the PowerQuicc II through a Media-Independent Interface (MII). It features :

- 10-Base-T and 100-Base-TX
- Auto-Negotiation and Parallel Detection
- MII interface with extended register capability
- Robust baseline wander correction performance
- Standard CSMA/CD or Full-Duplex operation
- MDIO management interface

Its management interface is controlled by the PowerQuicc ports PC(25) (MDC) and PC(26) (MDIO).

The LXT971A controls its own interrupt line to the local processor (-IRQ3).

The LXT971A reset input is controlled by the PowerQUICC II CPM I/O port PC(24) (0=reset).

The LXT971A also includes three programmable LED drivers, which are used to control the LEDs on the faceplate.

Table 1-29. Ethernet LEDs

LXT971 Output	Description
LED/CFG1	Faceplate LED 3 (green)
LED/CFG2	Faceplate LED 4 (green)
LED/CFG3	Faceplate LED 5 (green)

TDM Bus Configurations

General

The TDM bus general structures are described in Figure 1-6 for the general bus structure and in Figure 1-7 and Figure 1-8 for the general clock structure. This general structure allows three basic configurations that can each have several variants. The configurations are:

Direct Mode: The QuadFALC TDM busses are directly tied to the MPC8260. Two variants exist:

- Multiplex Direct Mode with one multiplexed TDM bus for the four framers. In that case the four framers have the same rhythm.
- Independent Direct Mode with one independent TDM bus per framer. Each framer can have its own rhythm which is the same in transmit and receive.

Switched Mode: The QuadFALC multiplexed TDM bus is tied to the first TDM bus on P4. The second TDM bus on P4 is tied to the MPC8260.

Pass Through Mode: Special mode that allows line snooping or concurrent treatment. It applies to framers 1 and 2 together and to framers 3 and 4 together.

The use and the source of each data and clock is described in the different mode descriptions.

The different modes are selected by programming the MPC8260 port PA(7) = SWMODE_N which selects the Switched mode and the port PA(0) = COMCLK_N which provides a common clock in Pass Through mode, by configuring the MPC8260 TDM ports (open drain output or high impedance input when unused) and by programming the QuadFALC registers.

Two network configurations are possible, Line Termination (LT) and Network Termination (NT).

In the LT configuration, the network synchronization comes from the lines: the QuadFALC derives its clocks from the receive rhythm and provides (directly in Direct mode or indirectly in Switched mode) synchronization for the TDM busses.

In the NT configuration, the card, considered as being part of the network, is master of the line rhythm. A network reference synchronization signal must be provided through PMC connector P4 in order to control the lines rhythm in accordance to the network. If this reference signal is not provided, or is temporarily failing, the card automatically provides a fixed frequency reference.

The framers description shown in this chapter is a partial description, please refer to the PEB22554 Data Sheet for a full description.

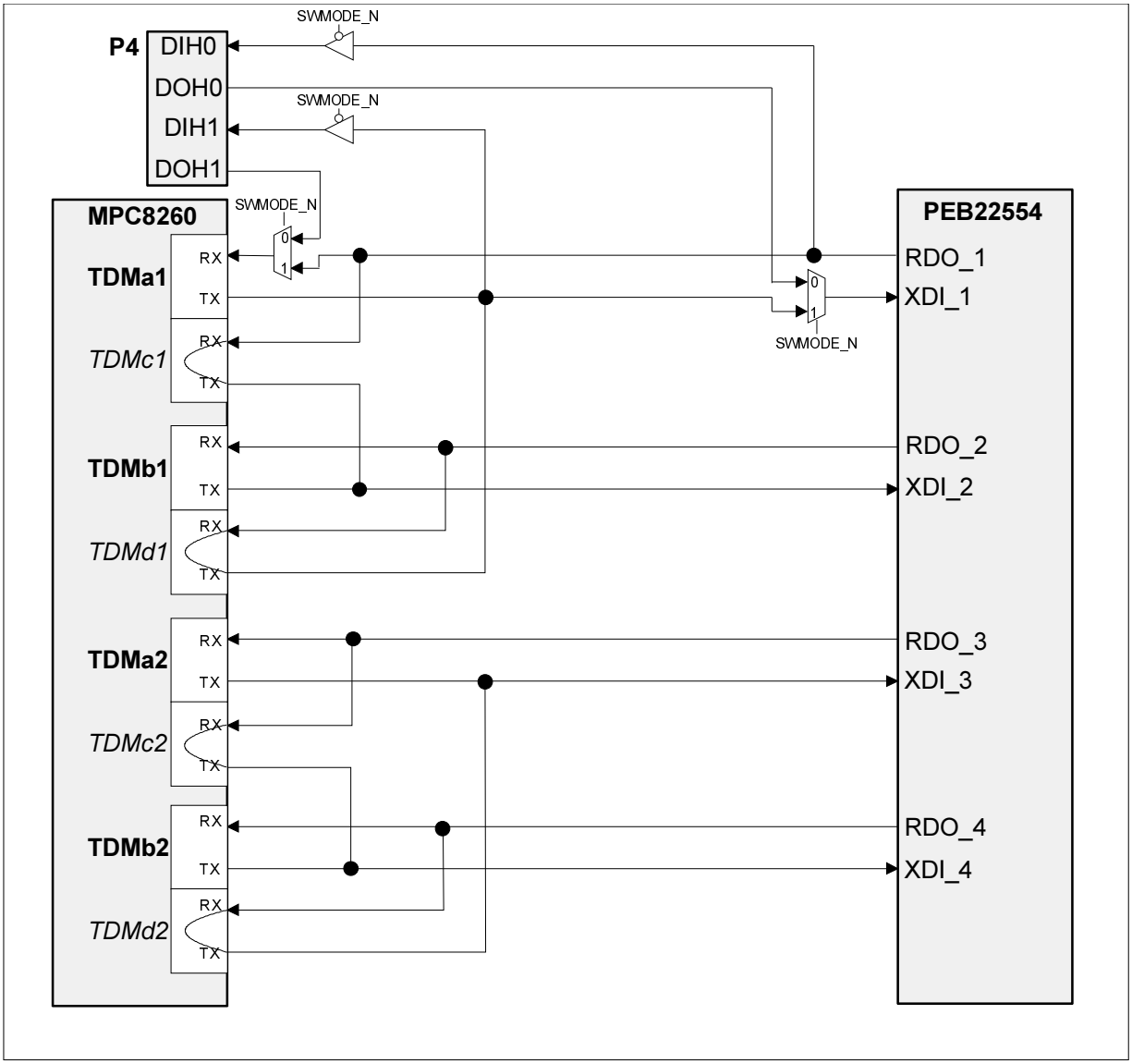


Figure 1-6. TDM Busses General Structure

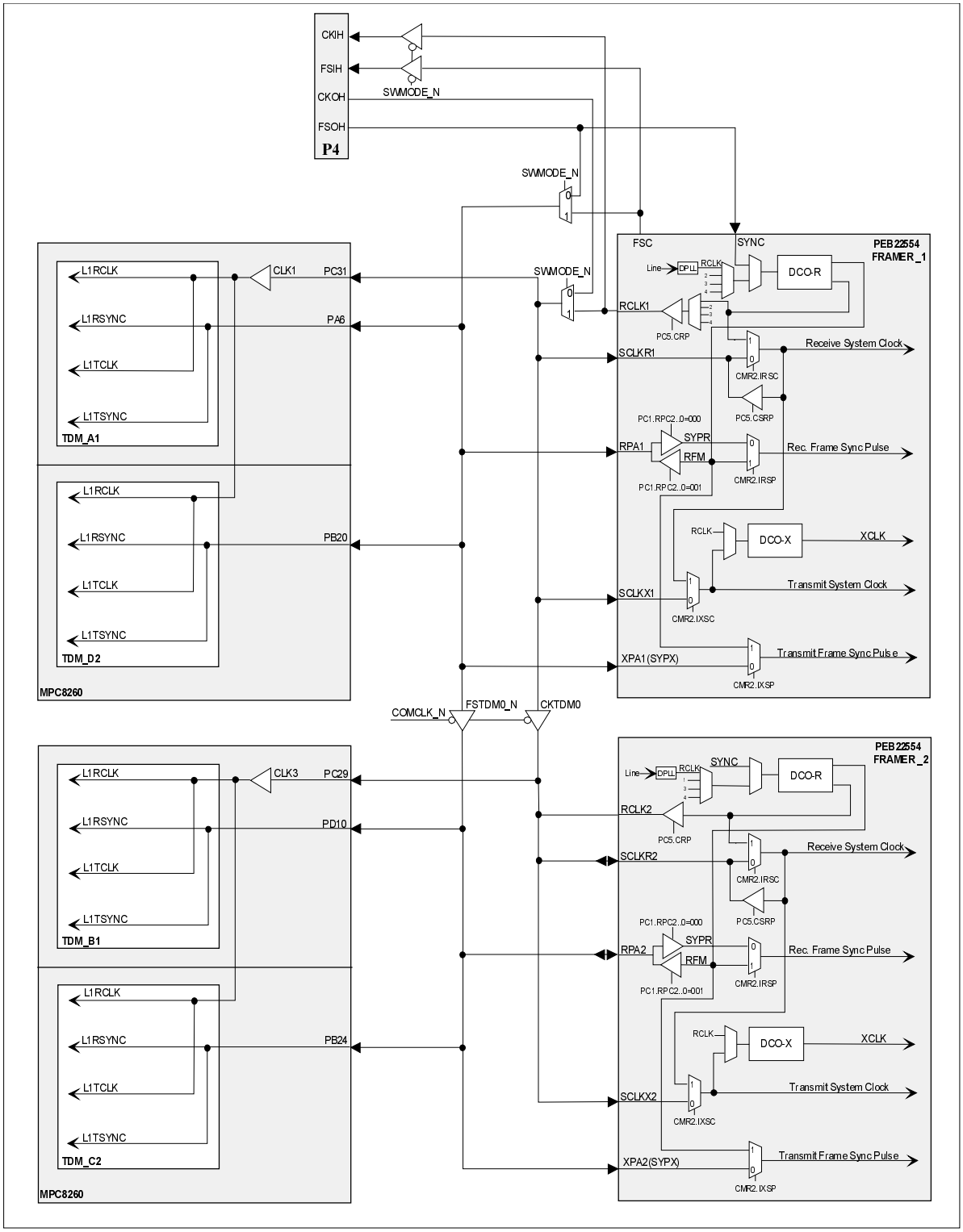


Figure 1-7. General Clock Structure (Framer 1 & 2)

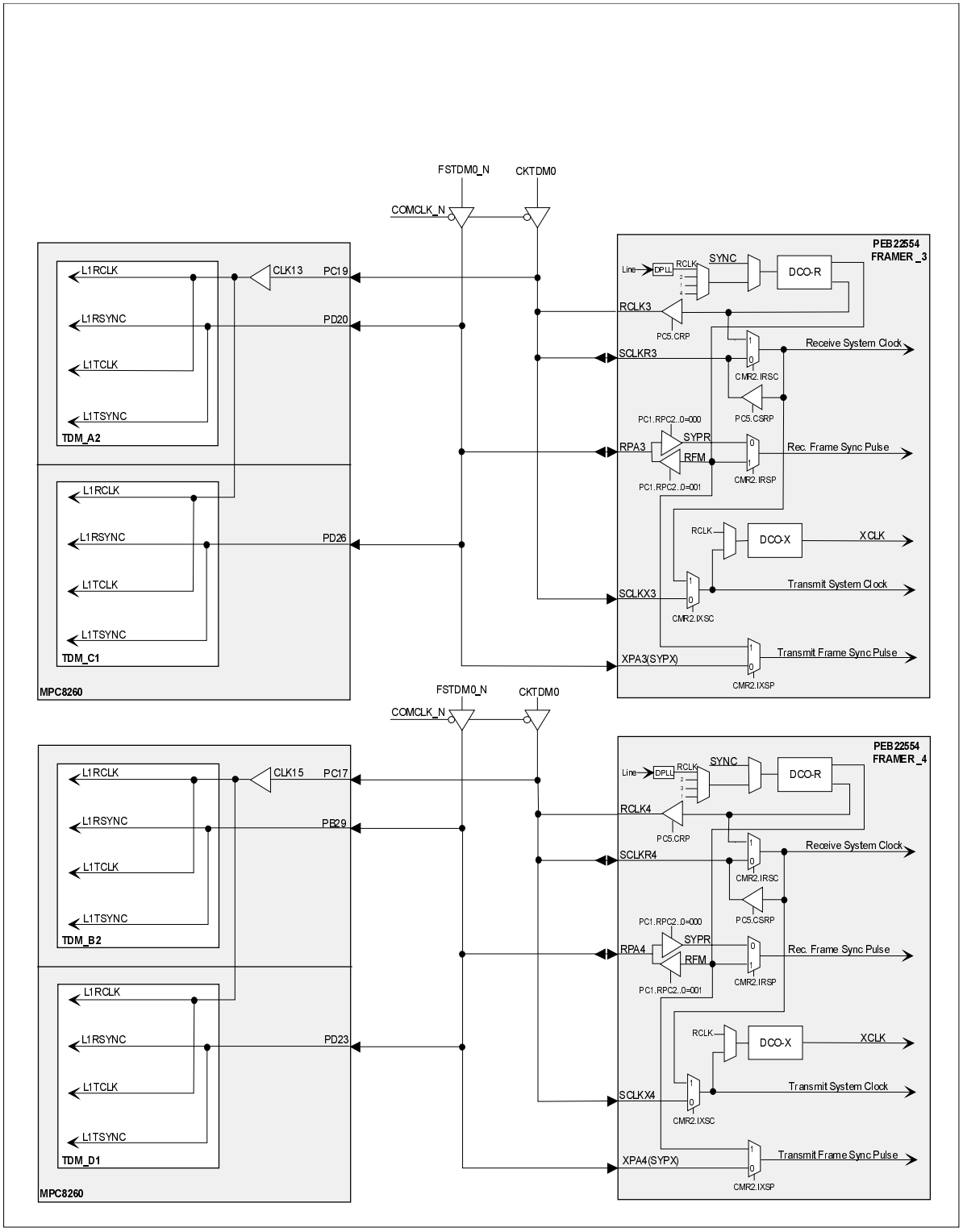


Figure 1-8. General Clock Structure (Framer 3 & 4)

Multiplex Direct Mode

In this mode, PA(7) = SWMODE_N = 1 and PA(0) = COMCLK_N = 1.

In multiplex direct mode, the four framers have the same rhythm. The QuadFALC system interface is in multiplex mode; the first QuadFALC TDM bus is directly tied to the CPM TDM bus TDMa1. The TDM bus clock and the frame synchronization signal are provided by the QuadFALC. In NT mode, the QuadFALC can synchronize on an external network reference clock provided on connector P4.

Figure 1-9, Figure 1-10, and Figure 1-11 show the specific implementation of this mode. Grey lines indicate unused connections.

Table 1-30. TDM and Synchronization Signals in Multiplex Direct Mode

Output	Input(s)	Description
RDO_1 (QuadFALC)	TDMa1_RX	8 Mb/s received data from the four E1/T1 lines. The QuadFALC system interface is in multiplex mode.
TDMa1_TX (CPM)	XDI_1	8 Mb/s transmit data for the four E1/T1 lines. The QuadFALC system interface is in multiplex mode.
FSC (QuadFALC)	RPA1, XPA1, TDMa1_L1RSYNC	8 KHz synchronization pulse generated by one of the four DCO-Rs, used for the TDM frame synchronization clock. RPA1 input is configured as $\overline{\text{SYPR}}$ and used for the Receive Frame Synchronous Pulse **(CMR2.IRSP=0)**. XPA1 is configured as $\overline{\text{SYPX}}$ and used for the Transmit Frame Synchronous Pulse **(CMR2.IXSP=0)**. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**.
RCLK1 (QuadFALC)	SCLKR1 SCLKX1 TDMa1_L1RCLK	8.192 MHz dejittered clock generated by the one of the four DCO-R circuits, output on RCLK1 **(PC5.CRP=1)** and used for the TDM bus clock. SCLKR1 input is used for the Receive System Clock **(CMR2.IRSC=0)**. SCLKX1 input is used for the Transmit System Clock **(CMR2.IXSC=0)** and provides the transmit rhythm to the DCO-X circuits. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**. In LT mode, DCO-R synchronizes on one of the four recovered line clocks. In NT mode, the DCO-R synchronizes on the external SYNC signal. When no reference clock is provided on SYNC, DCO-R is in free-running mode.
FSOH (P4)	SYNC	External synchronization clock provided to the DCO-R in NT mode. When no signal is provided, SYNC is tied to GND.

NOTE

TDMb1, TDMc1, TDMd1, TDMa2, TDMb2, TDMc2 and TDMd2 signals are not used and must be tristated.

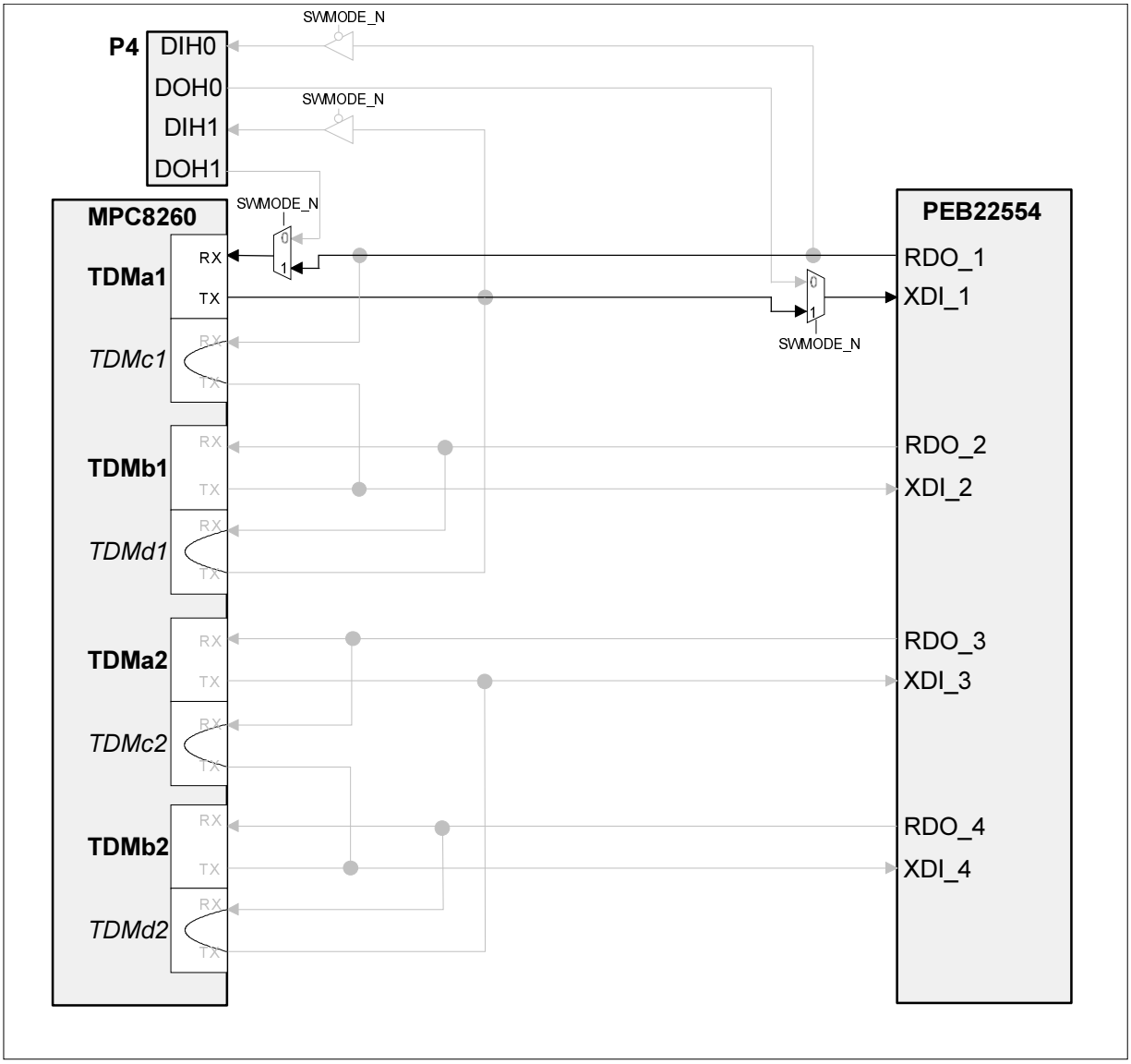


Figure 1-9. TDM Busses in Multiplex Direct Mode

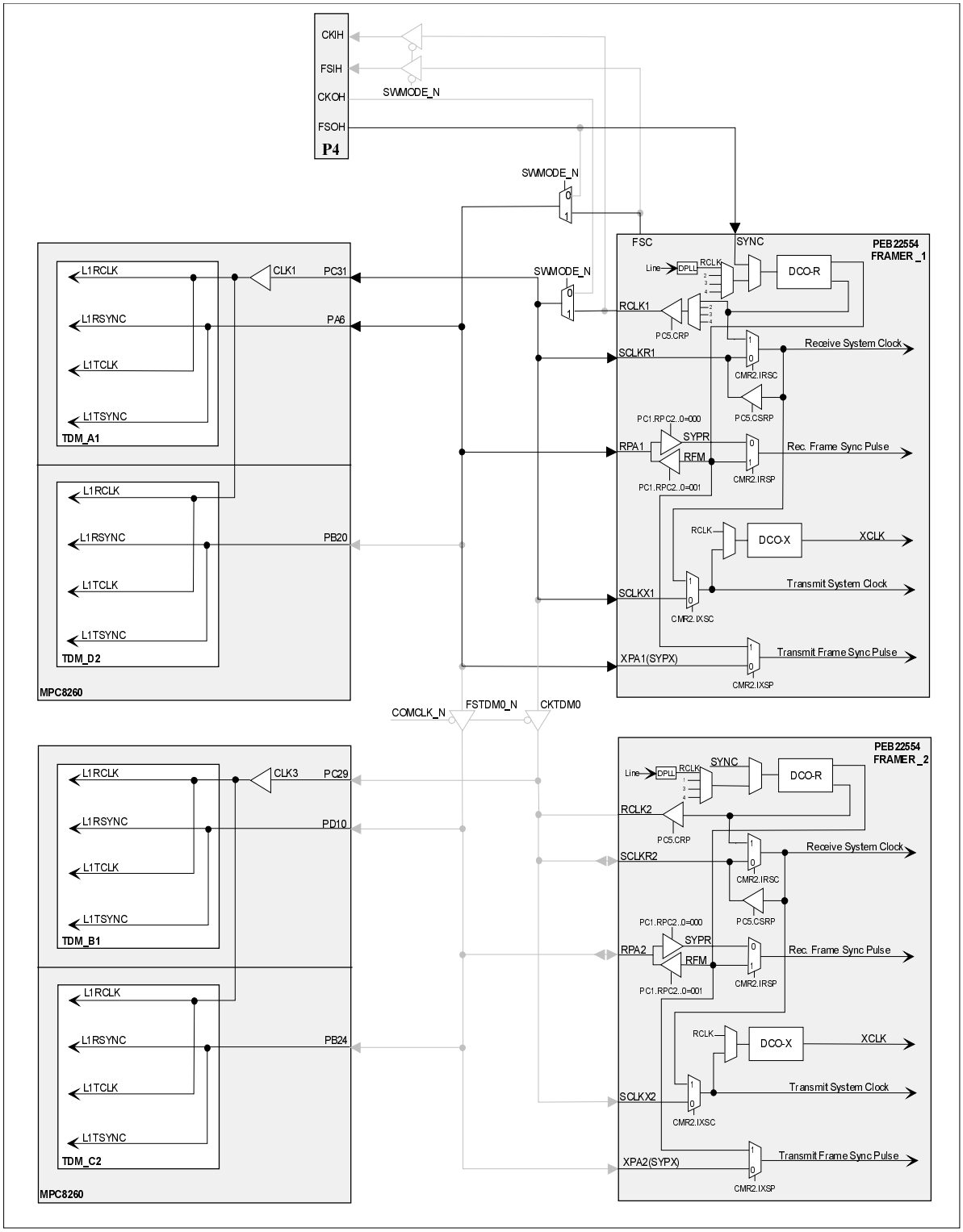


Figure 1-10. Clocks in Multiplex Direct Mode (Framer 1 & 2)

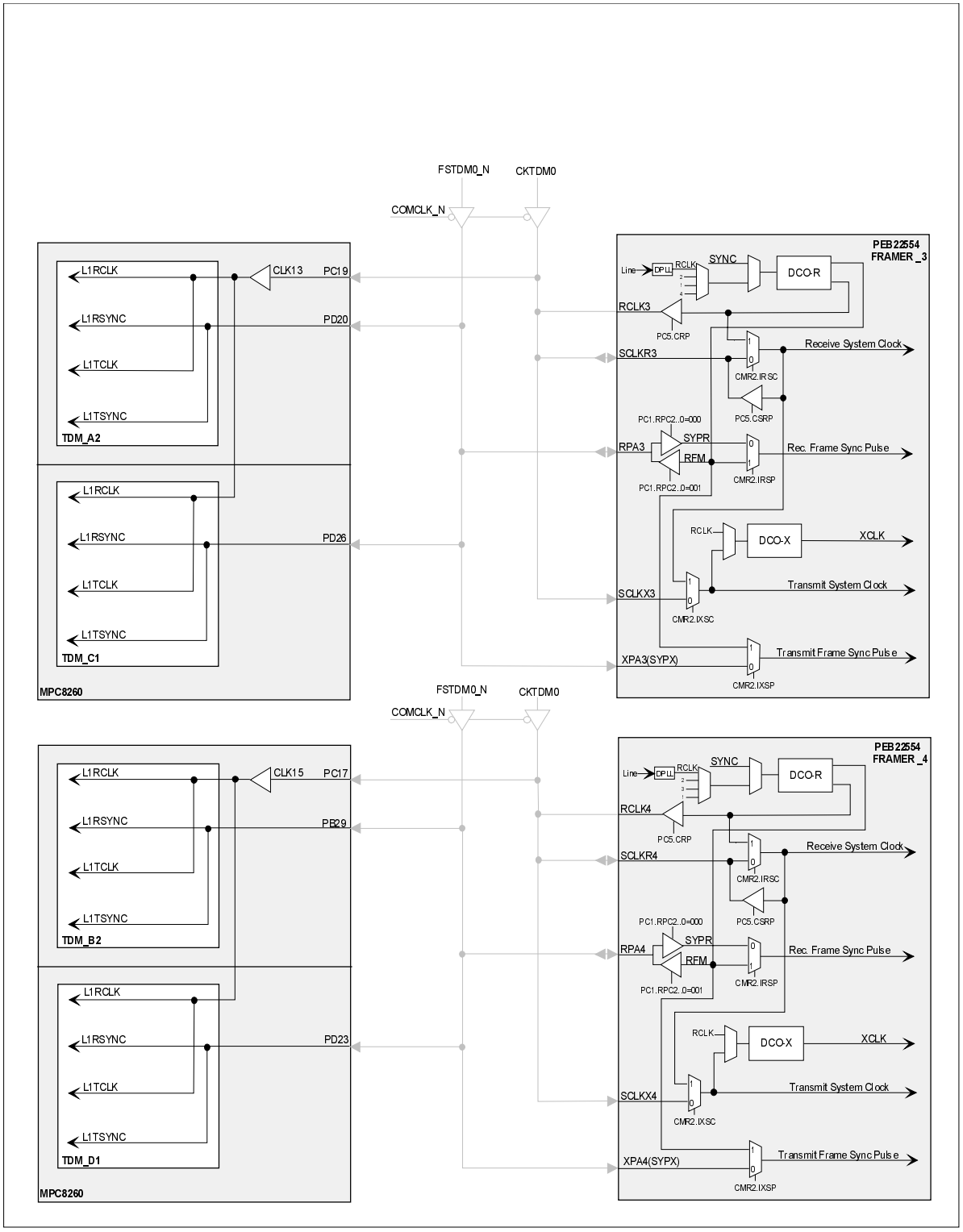


Figure 1-11. Clocks in Multiplex Direct Mode (Framer 3 & 4)

Independent Direct Mode

In this mode, PA(7) = SWMODE_N = 1 and PA(0) = COMCLK_N = 1.

In independent direct mode, each framer can have its own rhythm. Each QuadFALC TDM bus is directly tied to a CPM TDM bus and has its own clock and frame synchronization signal provided by the QuadFALC. In NT mode, each framer can synchronize on an external network reference clock provided on connector P4.

Figure 1-12, Figure 1-13, and Figure 1-14 show the specific implementation of this mode. Grey lines indicate unused connections.

Table 1-31. TDM and Synchronization Signals in Independent Direct Mode

Output	Input(s)	Description
RDO_1 (QuadFALC)	TDMa1_RX	2 Mb/s received data from the E1/T1 line.
TDMa1_TX (CPM)	XDI_1	2 Mb/s transmit data for the E1/T1 line.
FSC (QuadFALC)	RPA1, XPA1, TDMa1_L1RSYNC	8 KHz synchronization pulse generated by the DCO-R used for the TDM frame synchronization clock. RPA1 input is configured as $\overline{\text{SYPR}}$ and used for the Receive Frame Synchronous Pulse **(CMR2.IRSP=0)**. XPA1 is configured as $\overline{\text{SYPX}}$ and used for the Transmit Frame Synchronous Pulse **(CMR2.IXSP=0)**. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**.
RCLK1 (QuadFALC)	SCLKR1 SCLKX1 TDMa1_L1RCLK	2.048 MHz dejittered clock generated by the DCO-R circuit, output on RCLK1 **(PC5.CRP=1)** and used for the TDM bus clock. SCLKR1 input is used for the Receive System Clock **(CMR2.IRSC=0)**. SCLKX1 input is used for the Transmit System Clock **(CMR2.IXSC=0)** and provides the transmit rhythm to the DCO-X circuit. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**. In LT mode, the input of the DCO-R is the recovered line clock. In NT mode, the DCO-R synchronizes on the external SYNC signal. When no reference clock is provided on SYNC, DCO-R is in free running mode.
RDO_2 (QuadFALC)	TDMb1_RX	2 Mb/s received data from the E1/T1 line.
TDMb1_TX (CPM)	XDI_2	2 Mb/s transmit data for the E1/T1 line.
RPA2 (QuadFALC)	TDMb1_L1RSYNC	8 KHz frame synchronization pulse generated by the DCO-R and output on RPA2 configured as RFM. The Receive Frame Synchronous Pulse and the Transmit Frame Synchronous Pulse are internally generated, **(CMR2.IRSP=1, CMR2.IXSP=1)**. TDMb1 receive and transmit clocks are common **(SI1BMR.CRT=1)**.

Table 1-31. TDM and Synchronization Signals in Independent Direct Mode (cont)

Output	Input(s)	Description
SCLKR2 (QuadFALC)	TDMb1_L1RCLK	2.048 MHz dejittered Receive System Clock **(CMR2.IRSC=1)** generated by the DCO-R circuit, output on SCLKR2 **(PC5.CSRP=1)** and used for the TDM bus clock. The Transmit System Clock is sourced by the internal Receive System Clock **(CMR2.IXSC=1)** and provides the transmit rhythm to the DCO-X circuit. TDMb1 receive and transmit clocks are common **(SI1BMR.CRT=1)**. In LT mode, the input of the DCO-R is the recovered line clock. In NT mode, the DCO-R synchronizes on the external SYNC signal. When no reference clock is provided on SYNC, DCO-R is in free-running mode.
RDO_3 (QuadFALC)	TDMa2_RX	2 Mb/s received data from the E1/T1 line.
TDMa2_TX (CPM)	XDI_3	2 Mb/s transmit data for the E1/T1 line.
RPA3 (QuadFALC)	TDMa2_L1RSYNC	8 KHz frame synchronization pulse generated by the DCO-R and output on RPA3 configured as RFM. The Receive Frame Synchronous Pulse and the Transmit Frame Synchronous Pulse are internally generated, **(CMR2.IRSP=1, CMR2.IXSP=1)**. TDMa2 receive and transmit clocks are common **(SI2AMR.CRT=1)**.
SCLKR3 (QuadFALC)	TDMa2_L1RCLK	2.048 MHz dejittered Receive System Clock **(CMR2.IRSC=1)** generated by the DCO-R circuit, output on SCLKR3 **(PC5.CSRP=1)** and used for the TDM bus clock. The Transmit System Clock is sourced by the internal Receive System Clock **(CMR2.IXSC=1)** and provides the transmit rhythm to the DCO-X circuit. TDMa2 receive and transmit clocks are common **(SI2AMR.CRT=1)**. In LT mode, the input of the DCO-R is the recovered line clock. In NT mode, the DCO-R synchronizes on the external SYNC signal. When no reference clock is provided on SYNC, DCO-R is in free running mode.
RDO_4 (QuadFALC)	TDMb2_RX	2 Mb/s received data from the E1/T1 line.
TDMb2_TX (CPM)	XDI_4	2 Mb/s transmit data for the E1/T1 line.
RPA4 (QuadFALC)	TDMb2_L1RSYNC	8 KHz frame synchronization pulse generated by the DCO-R and output on RPA4 configured as RFM. The Receive Frame Synchronous Pulse and the Transmit Frame Synchronous Pulse are internally generated, **(CMR2.IRSP=1, CMR2.IXSP=1)**. TDMb2 receive and transmit clocks are common **(SI2BMR.CRT=1)**.

Table 1-31. TDM and Synchronization Signals in Independent Direct Mode (cont)

Output	Input(s)	Description
SCLKR4 (QuadFALC)	TDMb2_L1RCLK	2.048 MHz dejittered Receive System Clock **(CMR2.IRSC=1)** generated by the DCO-R circuit, output on SCLKR4 **(PC5.CSRP=1)** and used for the TDM bus clock. The Transmit System Clock is sourced by the internal Receive System Clock **(CMR2.IXSC=1)** and provides the transmit rhythm to the DCO-X circuit. TDMb2 receive and transmit clocks are common **(SI2BMR.CRT=1)**. In LT mode, the input of the DCO-R is the recovered line clock. In NT mode, the DCO-R synchronizes on the external SYNC signal. When no reference clock is provided on SYNC, DCO-R is in free running mode.
FSOH (P4)	SYNC	External synchronization clock provided to the DCO-R in NT mode. When no signal is provided, SYNC is tied to GND.

NOTES

- **RCLK2, RCLK3 and RCLK4 must be configured as inputs (PC5.CRP=0).**
- **XPA1, XPA2, XPA3 and XPA4 should be configured as $\overline{\text{SYPX}}$ (They must not be configured as outputs).**
- **TDMc1, TDMd1, TDMc2 and TDMd2 signals are not used and should be tristated.**

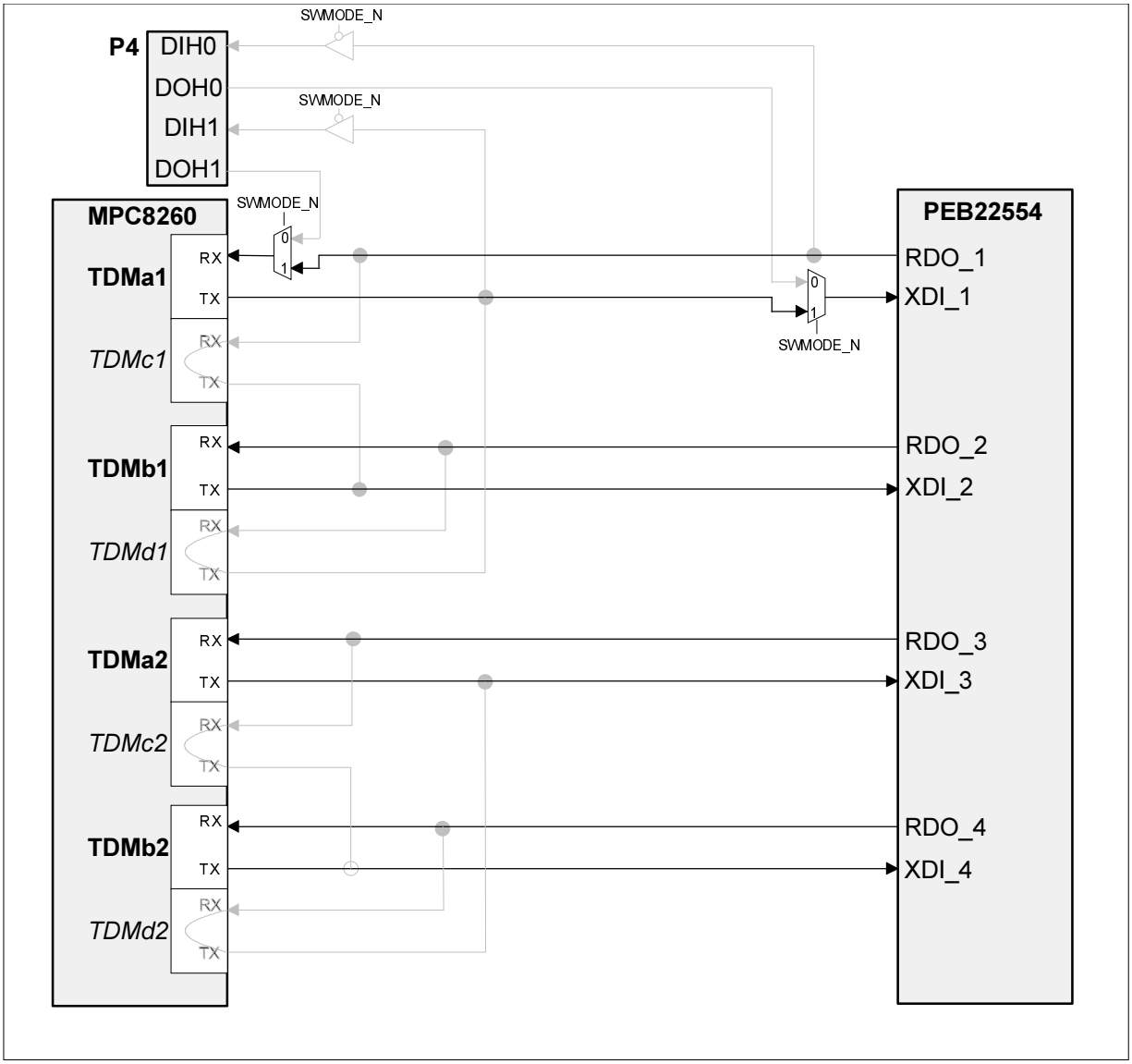


Figure 1-12. TDM Busses in Independent Direct Mode

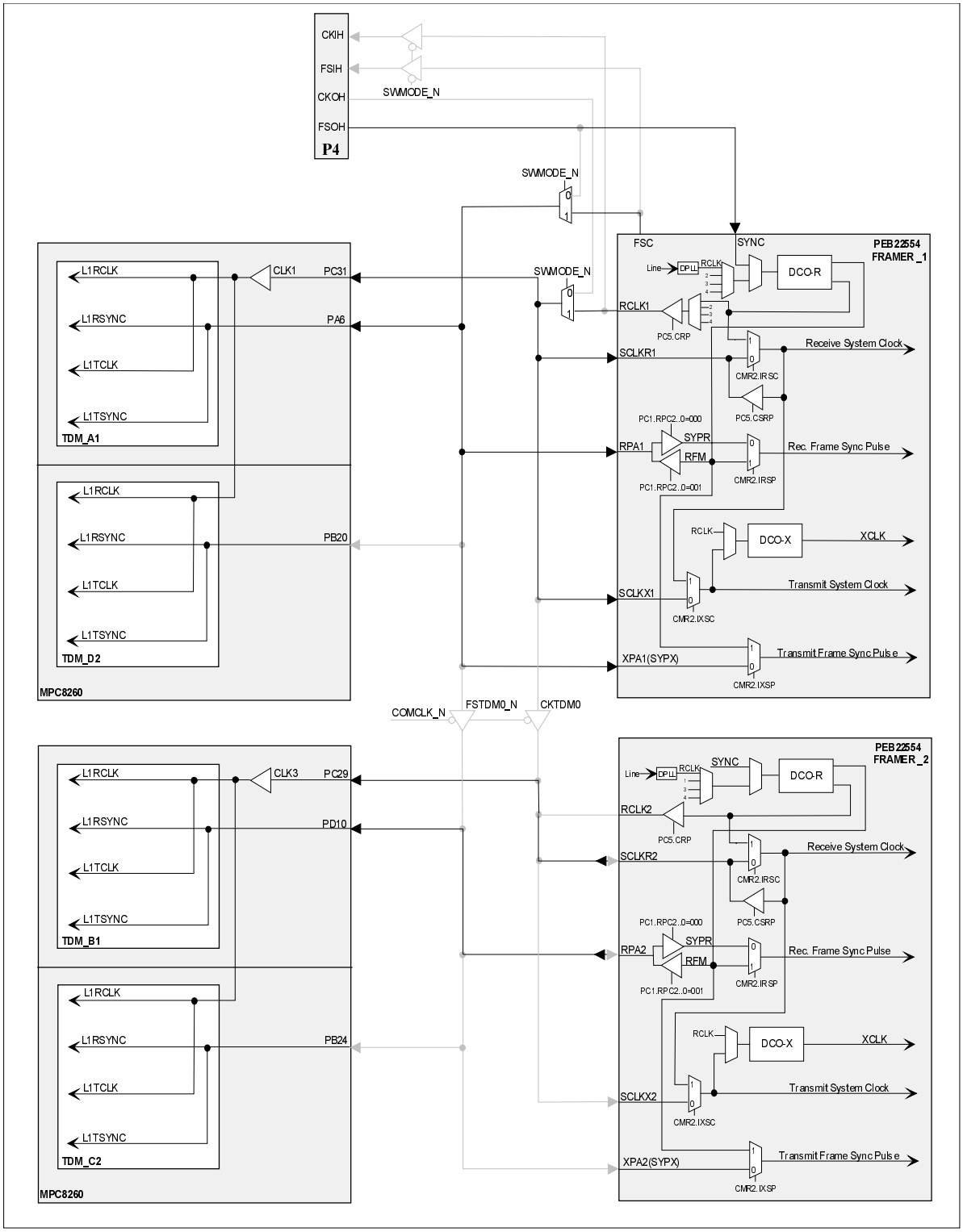


Figure 1-13. Clocks in Independent Direct Mode (Framer 1 & 2)

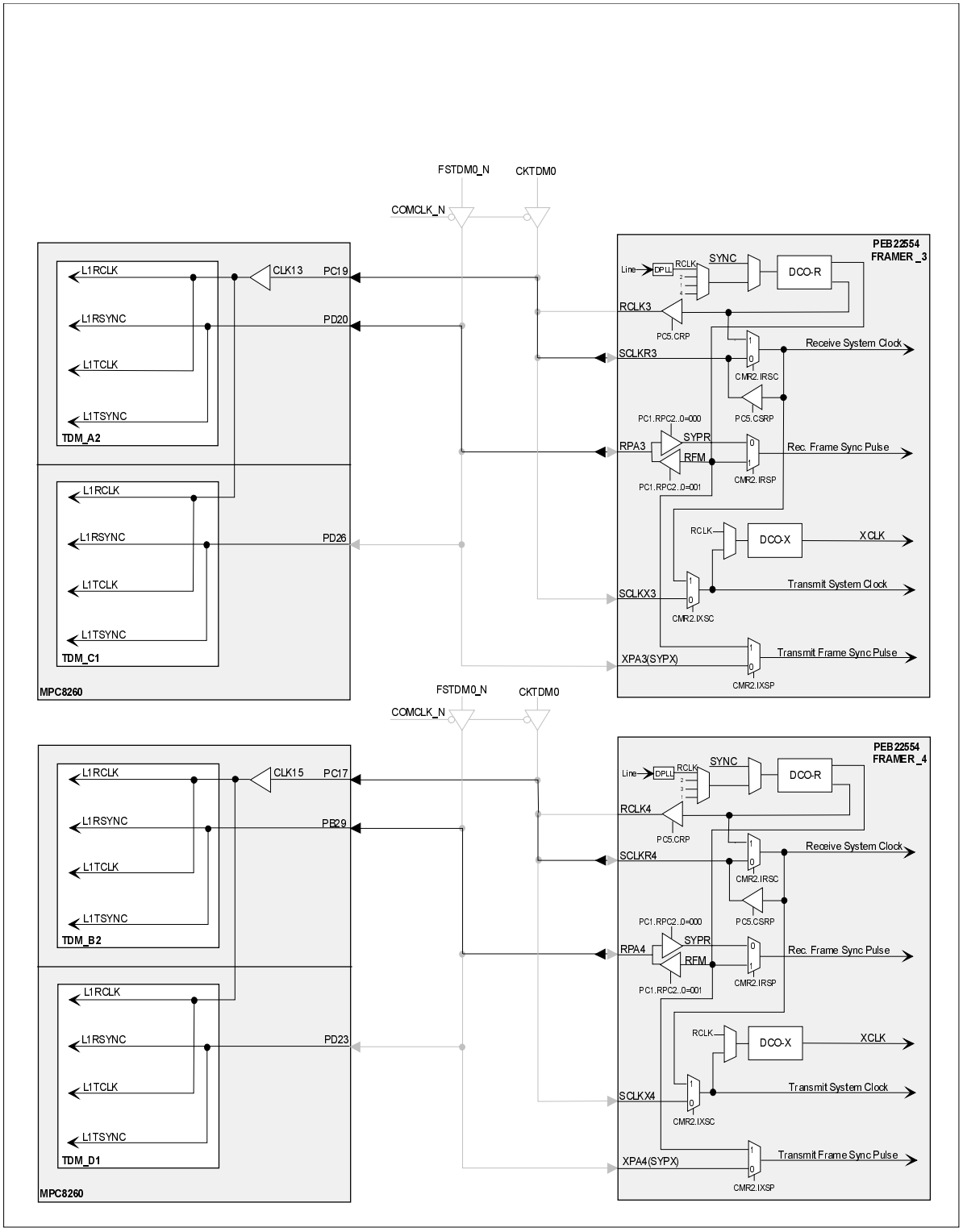


Figure 1-14. Clocks in Independent Direct Mode (Framer 3 & 4)

Switched Mode

In this mode, **PA(7) = SWMODE_N = 0** and PA(0) = COMCLK_N = 1.

In switched mode, the QuadFALC multiplexed TDM bus is tied to the first TDM bus on P4. The second TDM bus on P4 is tied to the MPC8260. The TDM busses clock and frame synchronization signals are provided by connector P4. In NT mode, the QuadFALC can synchronize on an external network reference clock provided on P4.

Figure 1-15, Figure 1-16 and Figure 1-17 show the specific implementation of this mode. Grey lines indicate unused connections.

Table 1-32. TDM and Synchronization Signals in Switched Mode

Output	Input(s)	Description
RDO_1 (QuadFALC)	DIH0	8 Mb/s received data from the four E1/T1 lines and sent to the first P4 TDM bus. The QuadFALC system interface is in multiplex mode.
DOH0 (P4)	XDI_1	8 Mb/s transmit data for the four E1/T1 lines from the first P4 TDM bus. QuadFALC system interface is in multiplex mode.
TDMa1_TX (CPM)	DIH1	8 Mb/s transmit data from the CPM TDMa1 bus to the second P4 TDM bus.
DOH1 (P4)	TDMa1_RX	8 Mb/s received data from the second P4 TDM bus to the CPM TDMa1 bus.
FSOH (P4)	TDMa1_L1RSYNC, RPA1, XPA1	8 KHz frame synchronization pulse provided by P4 for both TDM busses. RPA1 input is configured as $\overline{\text{SYPR}}$ and used for the Receive Frame Synchronous Pulse **(CMR2.IRSP=0)**. XPA1 is configured as $\overline{\text{SYPX}}$ and used for the Transmit Frame Synchronous Pulse **(CMR2.IXSP=0)**. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**. In LT mode, FSOH should be externally synchronized to the lines rhythm.
CKOH (P4)	TDMa1_L1RCLK, SCLKR1, SCLKX1	8.192 MHz clock provided by P4 for both TDM busses. SCLKR1 input is used for the Receive System Clock **(CMR2.IRSC=0)**. SCLKX1 input is used for the Transmit System Clock **(CMR2.IXSC=0)** and provides the transmit rhythm to the DCO-X circuits. TDMa1 receive and transmit clocks are common **(SI1AMR.CRT=1)**. In LT mode, CKOH should be externally synchronized to the lines rhythm. In NT mode, CKOH provides the line rhythm to the DCO-X circuit via SCLKX1.
FSC (QuadFALC)	FSIH	8 KHz synchronization pulse generated by the internal DCO1-R, synchronized to the lines and provided to P4.
RCLK1 (QuadFALC)	CKIH	Dejittered clock generated by the internal DCO1-R circuit, synchronized to the lines and provided to P4.

NOTE

TDMb1, TDMc1, TDMd1, TDMa2, TDMb2, TDMc2 and TDMd2 signals are not used and must be tristated.

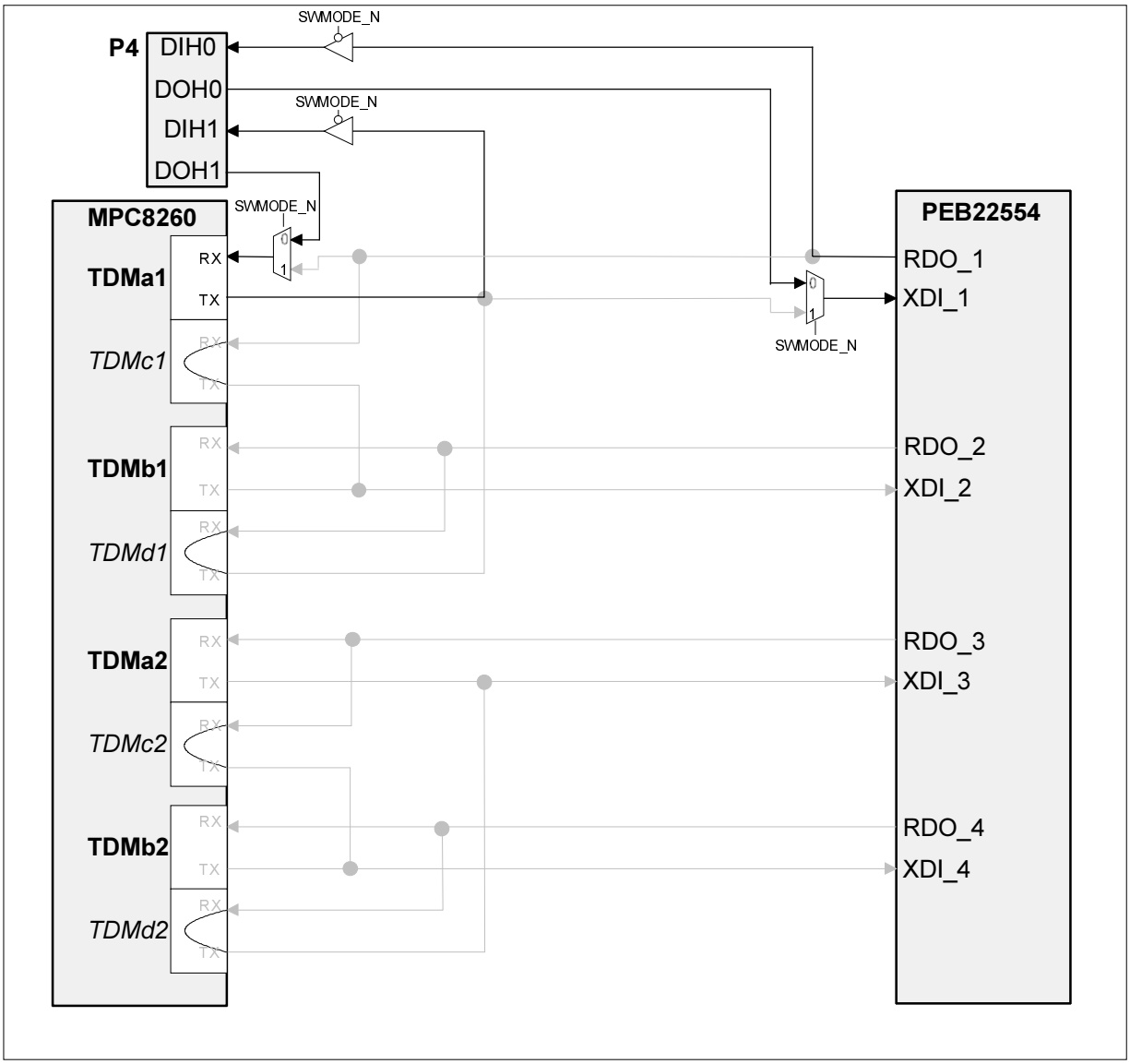


Figure 1-15. TDM Busses in Switched Mode

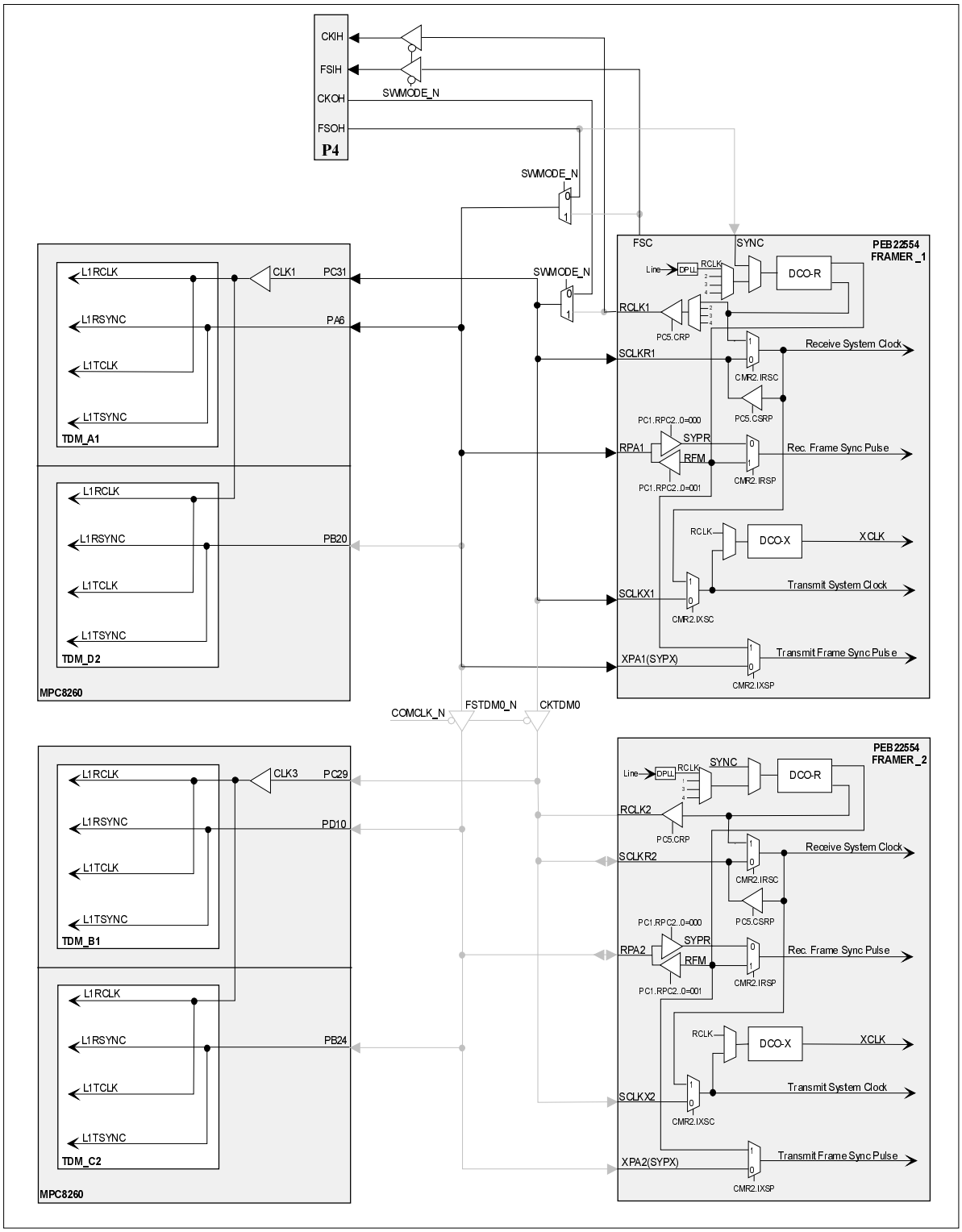


Figure 1-16. Clocks in Switched Mode (Framer 1 & 2)

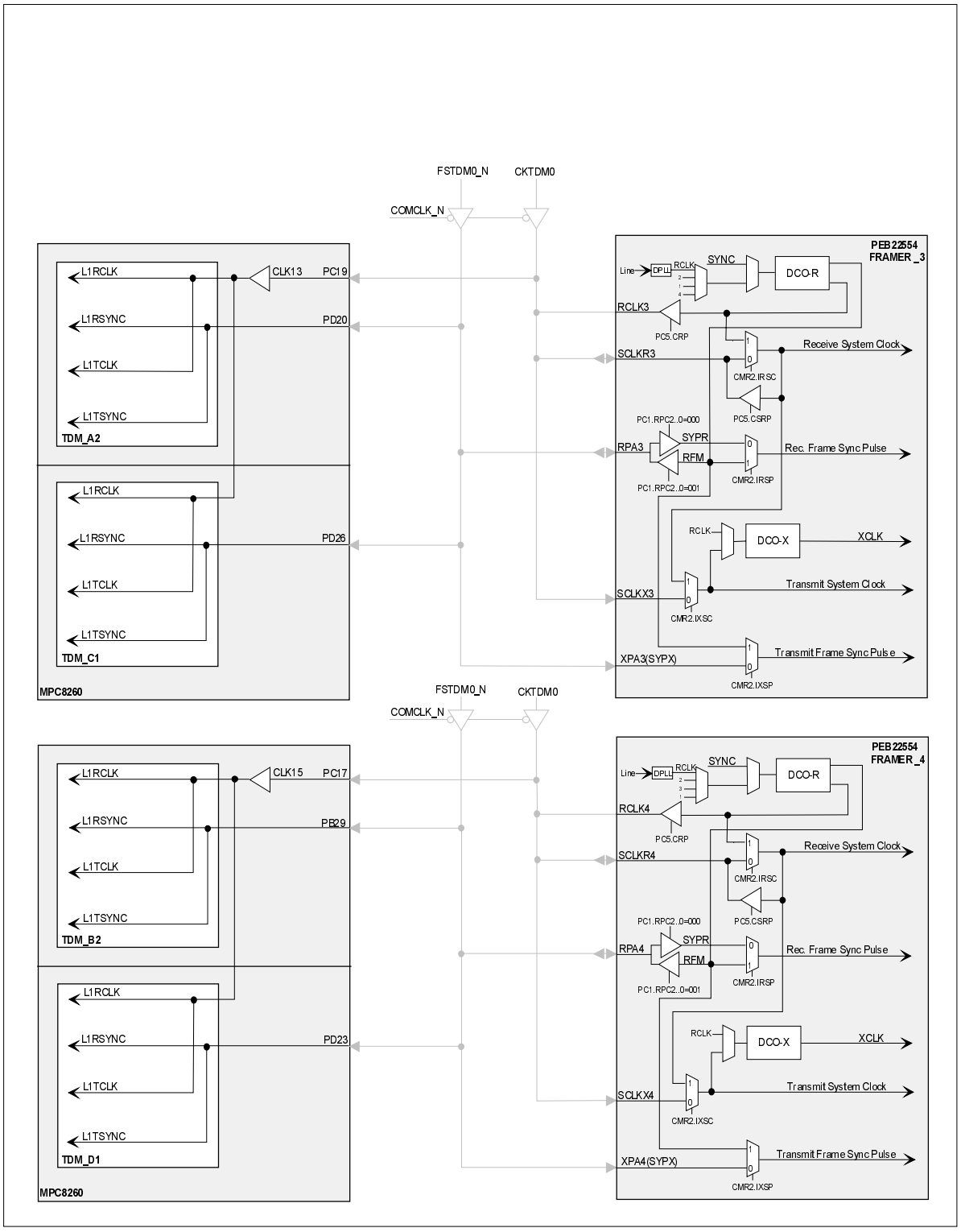


Figure 1-17. Clocks in Switched Mode (Framer 3 & 4)

Pass-Through Mode

In this mode, PA(7) = SWMODE_N = 1 and **PA(0) = COMCLK_N = 0**.

Pass through is possible from framer 1 to framer 2 and vice versa and from framer 3 to framer 4 and vice versa. The four framers have the same rhythm (**COMCLK_N = 0)**.

In framer 1 to framer 2 pass-through mode, the first framer is tied to the network in LT mode. Data received from this framer goes to TDMa1 and to the second framer, which is in NT mode, by using TDMc1 in Echo Mode, so that it can be connected to another adapter configured as a line Termination (LT) circuit. Data received from framer 2 is combined with data from TDMa1 by using TDMd1 in Echo Mode, and sent by framer 1; **TDMd1_TX and TDMa1_TX must be configured as open drain ports**. TDMb1 is not used and must be configured as input. Framer 2 to framer 1 pass-through description is symmetrical. The same description applies to framer 3 and framer 4.

Figure 1-18, Figure 1-19, Figure 1-20 and Figure 1-21show the specific implementation of this mode. Only the framer 1 to framer 2 and framer 3 to framer 4 pass through is described. Grey lines indicate unused connections.

Table 1-33. TDM and Synchronization Signals in Pass Through Mode

Output	Input(s)	Description
RDO_1 (QuadFALC)	TDMa1_RX, TDMc1_RX	2 Mb/s received data from the 1st E1/T1 line and sent to the 2nd line via TDMc1 in echo mode.
RDO_2 (QuadFALC)	TDMd1_RX	2 Mb/s received data from the 2nd E1/T1 line and sent to the 1st line via TDMd1 in echo mode. This data will be completed (anded) with data from TDMa1_TX.
RDO_3 (QuadFALC)	TDMa2_RX, TDMc2_RX	2 Mb/s received data from the 3rd E1/T1 line and sent to the 4th line via TDMc2 in echo mode.
RDO_4 (QuadFALC)	TDMd2_RX	2 Mb/s received data from the 4th E1/T1 line and sent to the 3rd line via TDMd2 in echo mode. This data will be completed (anded) with data from TDMa2_TX.
TDMa1_TX (CPM) and TDMd1_TX (CPM)	XDI_1	2 Mb/s transmit data for the 1st E1/T1 line. Data from TDMa1_TX and TDMd1_TX are anded **(ports are open-drain outputs)**.
TDMc1_TX (CPM)	XDI_2	2 Mb/s transmit data from the 1st E1/T1 line to the 2nd line. TDMc1 is in echo mode.
TDMa2_TX (CPM) and TDMd2_TX (CPM)	XDI_3	2 Mb/s transmit data for the 3rd E1/T1 line. Data from TDMa2_TX and TDMd2_TX are anded **(ports are open drain outputs)**.
TDMc2_TX (CPM)	XDI_4	2 Mb/s transmit data from the 3rd E1/T1 line to the 4th line. TDMc2 is in echo mode.

Table 1-33. TDM and Synchronization Signals in Pass Through Mode (cont)

Output	Input(s)	Description
FSC (QuadFALC)	RPA1, XPA1, RPA2, XPA2, RPA3, XPA3, RPA4, XPA4, TDMa1_L1RSYNC TDMb1_L1RSYNC TDMc1_L1RSYNC TDMd1_L1RSYNC TDMa2_L1RSYNC TDMb2_L1RSYNC TDMc2_L1RSYNC TDMd2_L1RSYNC	8 KHz synchronization pulse generated by the DCO1-R used for the TDM frame synchronization clocks. RPA1, 2, 3, 4 inputs are configured as $\overline{\text{SYPR}}$ and used for the Receive Frame Synchronous Pulse **(CMR2.IRSP=0)**. XPA1, 2, 3, 4 are configured as $\overline{\text{SYPX}}$ and used for the Transmit Frame Synchronous Pulse **(CMR2.IXSP=0)**. TDMs receive and transmit clocks are common **(SIxxMR.CRT=1)**.
RCLK1 (QuadFALC)	SCLKR1, SCLKX1, SCLKR2, SCLKX2, SCLKR3, SCLKX3, SCLKR4, SCLKX4, TDMa1_L1RCLK, TDMb1_L1RCLK, TDMa2_L1RCLK, TDMb2_L1RCLK	Internal 2.048 MHz dejittered receive clock generated by the DCO1-R circuit, output on RCLK1 **(PC5.CRP=1)** and used for the TDM bus clocks. SCLKR1, 2, 3, 4 inputs are used for the Receive System Clock **(CMR2.IRSC=0)**. SCLKX1,2,3,4 inputs are used for the Transmit System Clock **(CMR2.IXSC=0)** and provides the transmit rhythm to the DCO-X circuits. TDMa1, b1, a2, b2 receive and transmit clocks are common **(SIxxMR.CRT=1)**. TDMd2, c2, c1, d1 also use this clock for receive and transmit. The input of the DCO1-R is one of the four recovered line clocks.

NOTE

Unused TDM signals must be tristated.

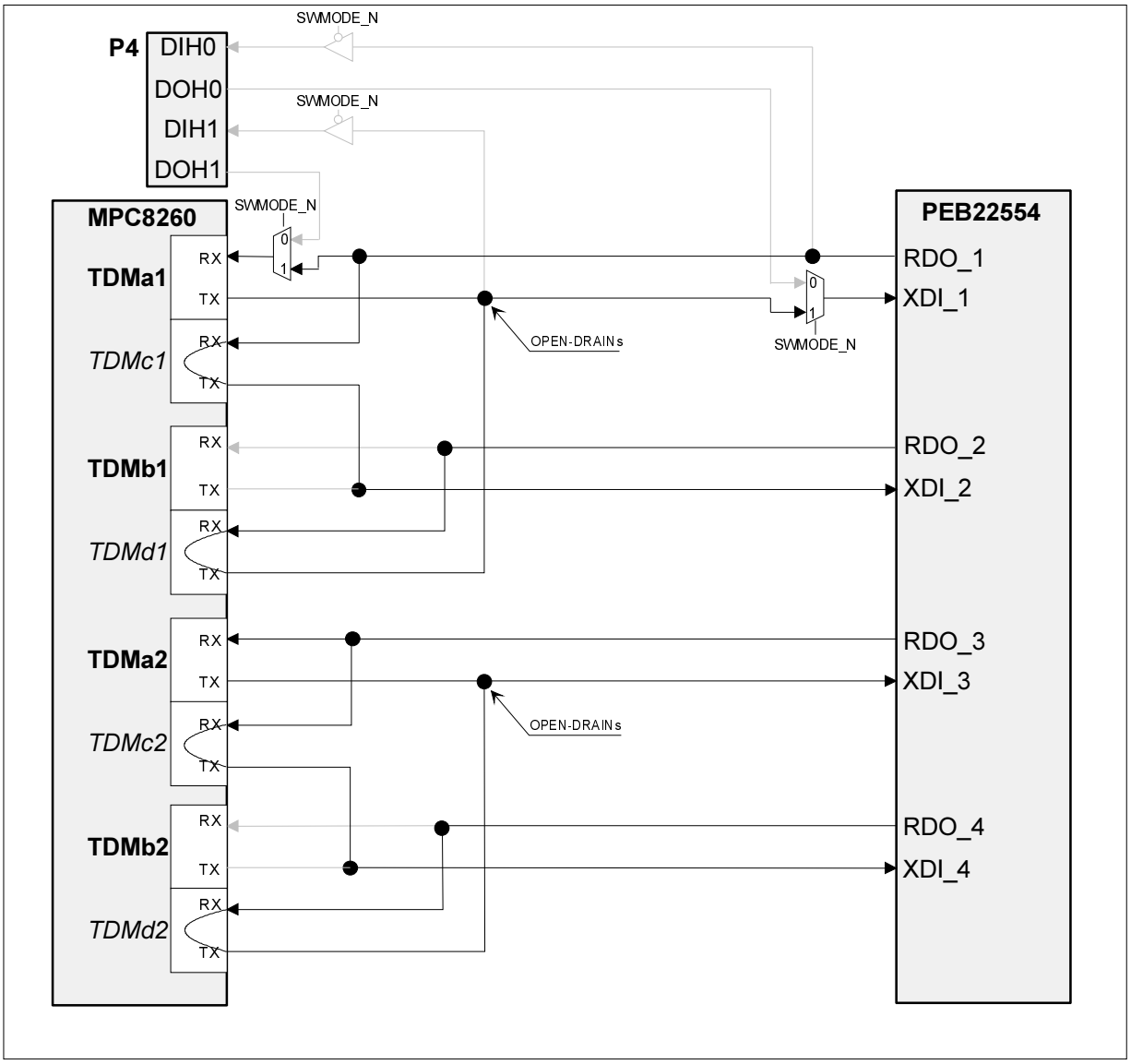


Figure 1-18. TDM Busses in Pass-Through Mode (1->2 & 3->4 Example)

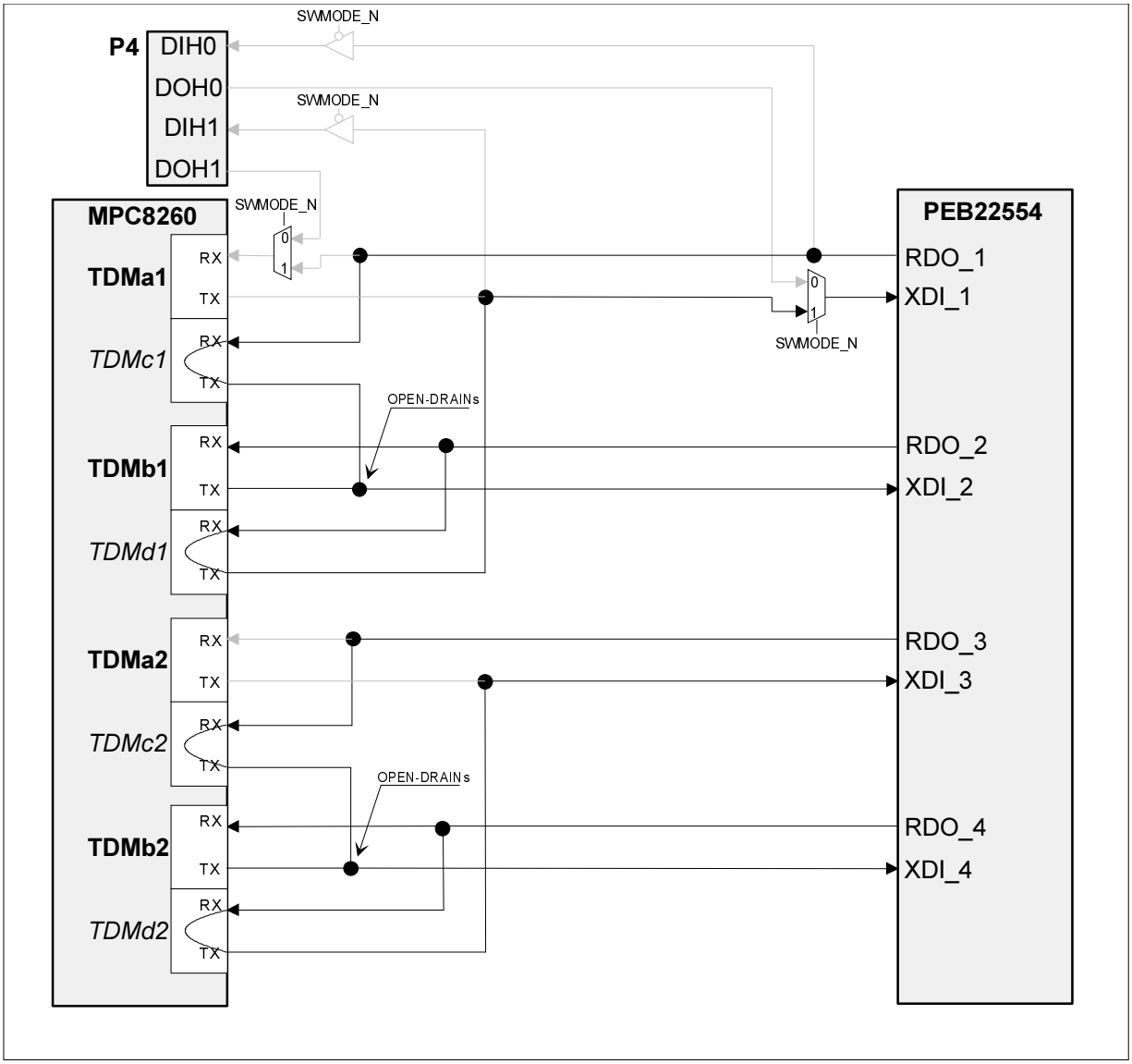


Figure 1-19. TDM Busses in Pass-Through Mode (2->1 & 4->3 Example)

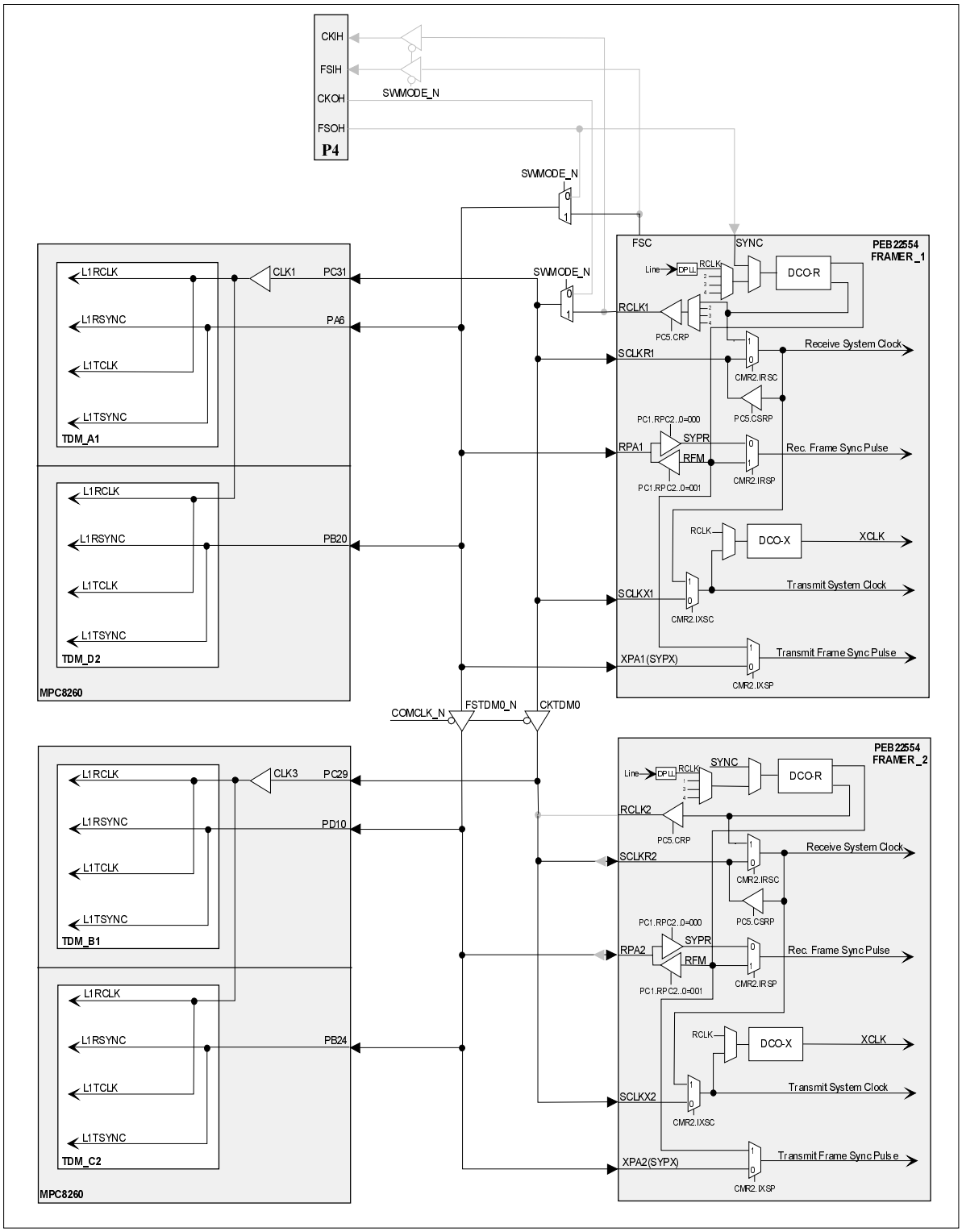


Figure 1-20. Clocks in Pass-Through Mode (Framer 1 & 2)

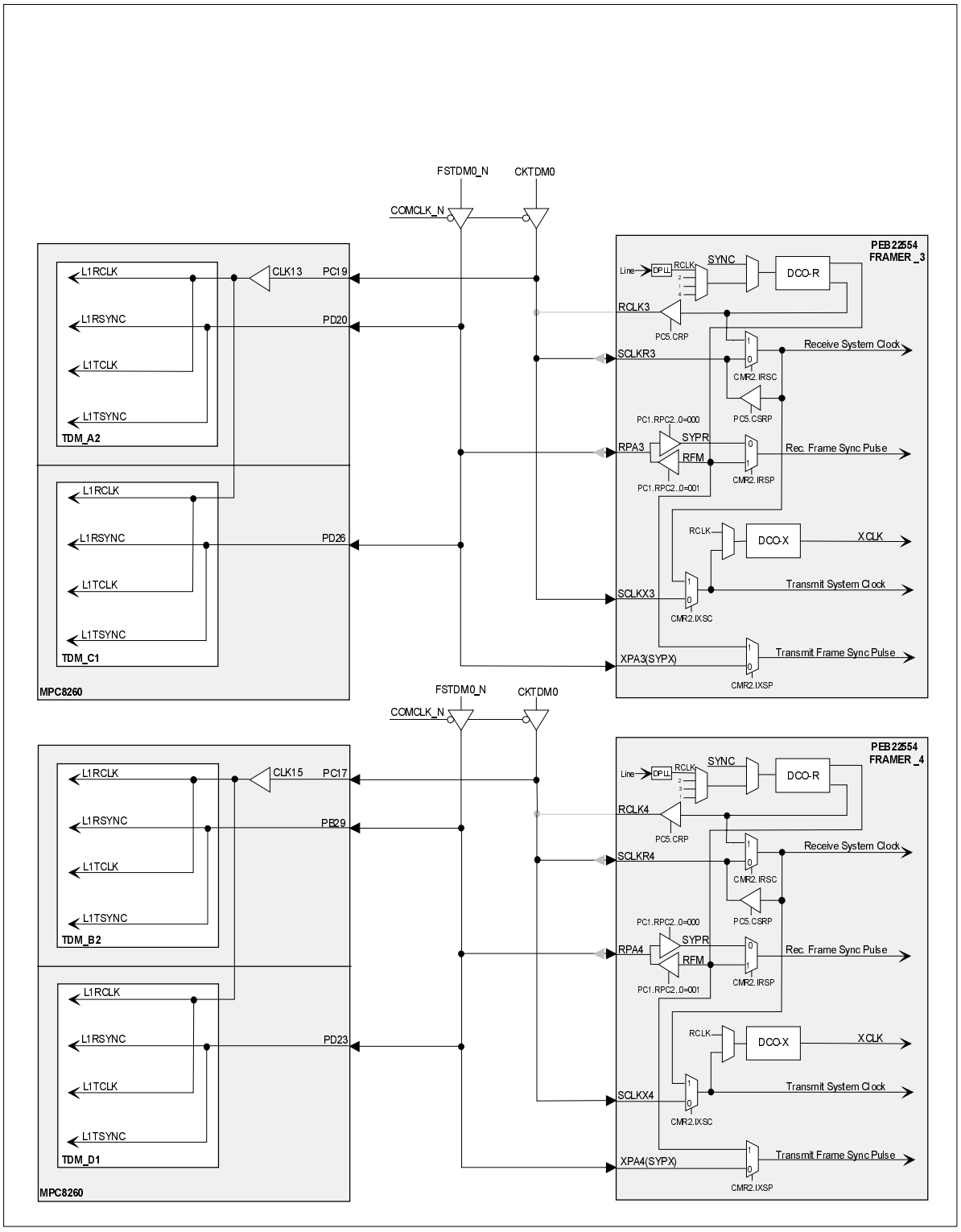


Figure 1-21. Clocks in Pass-Through Mode (Framer 3 & 4)

4538 Power-Up Initialization

2

Overview

After power-up, the STARTUP code is executed. This code is written entirely in assembly language and is the entry point after a power-up or a reset exception. STARTUP configures the PowerQUICC II and several other critical hardware elements such as the SDRAM memories. Once STARTUP is executed, code written in a high-level language such as “C” is executed.

This chapter describes this STARTUP initialization.

NOTE

The STARTUP source code is provided in the `APP/ASM/STARTUP.ASM` file.

PowerSpan Initialization

The PowerSpan is initialized by several different mechanisms.

During the power-up phase, the PowerSpan uses some of its I/O pins to determine its Hardware Configuration Word (also called Power-up options in PowerSpan documentation).

Then the PowerSpan initializes several of its internal registers by loading their values from the I²C serial EEPROM. Among these registers, the PCI vendor and device identification, the size of the PCI-to-Local windows, and the size and position of the first Local to PCI window are initialized by the serial EEPROM.

The addresses in the PCI space of the PCI-to-Local windows are chosen by the PCI host during its boot and programmed in the PCI configuration registers.

All the other initializations must be done either by the PowerSpan during its boot or by the PCI host.

PowerSpan Hardware Configuration Word

On the board, this configuration is defined as follows:

- PB_ARB_EN=0: Disable PowerSpan arbiter for 60x local bus
- P1_ARB_EN=0: Disable PowerSpan arbiter for PCI 1 bus
- P2_ARB_EN=0: Disable PowerSpan arbiter for PCI 2 bus (no PCI 2 bus)
- PWRUP_PRI_PCI=0: PCI 1 is primary PCI bus
- P1_R64_EN=0: Disable PCI 1 REQ64

- PWRUP_BOOT=0: The PowerQUICC II boots locally (not through PCI)
- PWRUP_DEBUG_EN=0: Disable debug mode
- PWRUP_BYPASS_EN=0: Disable PLL bypass

PowerSpan Register Initialization Through the I²C Serial EEPROM

Table 2-1 provides the PowerSpan Register initialization values stored in the Serial EEPROM. Refer to PowerSpan documentation, section *EEPROM Loading* for detailed mapping between EEPROM addresses and PowerSpan registers.

.

Table 2-1. PowerSpan Register Initialization Values in the Serial EEPROM

EEPROM Address	Initialization Value	Description
0	0x02	Long EEPROM load
5	0x0C	PCI Bus Master Enable and PCI memory Space Enable
6	0x00	PCI target Image prefetch indicators (all 0)
7–8	0x9070	PCI subsystem device ID
0x09–0x0A	0x107E	PCI subsystem vendor ID
0xB	0x02	Interrupt pin, INTA# is used on the PCI bus.
0xC	0x1C	PCI Base Address configuration registers enabled for the PowerSpan registers and target Image 0 and 1 (two PCI-to-local windows). Target images 1, 2, and 3 not enabled.
0xD	0x53	PCI target Image 0 = 2 MB, PCI target image 1 = 512 KB
0xE	0x00	PCI target images 2 and 3: size not defined (not enabled)
0xF	0x00	PCI Vital Product Data disabled
0x10	0x00	No PCI Lockout
0x11	0x5F	Interrupt pins direction: all outputs except –INT5.
0x12	0x00	PCI I²O target image disabled
0x13–0x1F	0x00	(Reserved)
0x20–0x21	0x9070	PCI device ID
0x22–0x23	0x107E	PCI vendor ID
0x24	0x02	PCI Base Class Code: Network Controller
0x25	0x80	PCI Sub Class Code: Other
0x26	0x00	PCI programming Interface
0x27	0x01	Revision register
0x28–0x30	0x00	PB slave image 0 disabled (local to PCI window)
0x31–0x33	0xF00200	PB slave register image base address = 0xF0020000

Table 2-1. PowerSpan Register Initialization Values in the Serial EEPROM (cont)

EEPROM Address	Initialization Value	Description
0x34–0x3F	0x00	(Reserved)

Other PowerSpan Initializations

It is necessary to initialize the PowerSpan Interrupt Map registers in a specific way, in order to use the interrupt pins as specified for the 4538. This can be done by the local processor during its boot and/or by the PCI host.

The following C code is an example of interrupt pin initialization.

Example 2-1. PowerSpan Interrupt Map Registers Initialization Code

```
ConfigurePspanInterrupts( void)
{
    RegWrite32( T_IER0,     0x00000000); // No interrupt source enabled
    RegWrite32( T_IER1,     0x00000000); // No interrupt source enabled

    /* Map Interrupt sources to Interrupt pins (INT0-INT5 for I/O usage) */

    RegWrite32( T_IDR,      0x5F000000); // Interrupt dir: -INT5 only as input
    RegWrite32( T_IMR_MBOX, 0x44440000); // MBOX0-3 to -INTA, MBOX4-7 to -INT0
    RegWrite32( T_IMR_DB,   0xECA86420); // DB0 to -INTA, DB2:7 to -INT0:5
    RegWrite32( T_IMR_DMA,  0x00004400); // DMAX0-1 to -INTA, DMA2-3 to -INT0
    RegWrite32( T_IMR_HW,   0x02ECA864); // No map from one int pin to an
other
    RegWrite32( T_IMR_P1,   0x00000000); // PCI 1 errors on PCI 1 -INTA
    RegWrite32( T_IMR_PB,   0x22222222); // PB errors on -INT0
    RegWrite32( T_IMR2_PB,  0x22200000); // PB errors on -INT0
    RegWrite32( T_IMR_MISC, 0x20000000); // I2O host to -INTA, I2O IOP to -
INT0
    RegWrite32( T_IER0,     0x001F0000); // INT0-4 HW interrupts enabled
}
```

PowerQUICC II Hardware Configuration Word

When the PowerQUICC II hardware reset signal is de-asserted, the PowerQUICC II generates 64-bit reads into its boot memory (the FLASH) with addresses starting at 0 and incremented by 8. The first eight bytes set its Hard Reset Configuration.

For the 4538, the PowerQUICC II Hard Reset Configuration is (must be):

- EARB = 0: Internal bus arbitration
- EXMC = 0: The internal memory controller is used
- CDIS = 0: The core is active
- EBM = 1: 60x-compatible bus mode
- BPS = 01: 8-bit boot port size
- –CIP = 0: Initial vector table base address is 0xFFF0 0000
- ISPS = 0: Responds as 64-bit slave to 64-bit masters
- L2CPC = 10: L2 cache pins configured as BADDR
- DPPC = 00: Data parity pins used for interrupt signals IRQ1–7
- ISB = 110: Internal Memory Mapped Register base address is 0xFF00 0000
- BMS = 0: Boot memory space is 0xFE00 0000
- BBD = 0: Bus Busy pins are enabled

- MMR =11: External bus requests are masked (PQ2 is the boot master)
- LBPC = 00: Local bus enabled
- APPC = 10: Address parity pins used for bank select
- CS10PC =01: –CS10/–BCTL1 used as –BCTL1
- MODCK_H =0101: PLL multiplication factors: with MODCK[1:3]=111, Bus @66, CPM @133, Core @200 MHz

PowerQUICC II Initializations

After a power-up or a reset exception, the PowerQUICC II must initialize itself and adapt its System Interface Unit (SIU) to the 4538 hardware. It must set up its memory controllers and Chip Selects. Then it must also initialize the SDRAM devices, before using them as its system memory.

PowerQUICC II System Interface Unit (SIU) Initialization

The PowerQUICC II SIU includes the following elements:

- System configuration and protection
- System reset monitoring and generation
- Clock synthesizer
- Power management
- 60x bus interface
- Memory Control Units

Several registers of the SIU need to be initialized during boot time for proper operation.

Internal Memory Map Register (IMMR)

The PowerQUICC II IMMR register is normally properly set in the Reset Configuration Word to map the PowerQUICC II Internal registers to address 0xFF010000.

Bus Configuration Register (BCR)

Some fields of the BCR register are initialized by the Reset Configuration Word. Several other fields however, need to be initialized:

- EBM = 1: 60x bus mode
- APD = 010: Wait two cycles for ARTRY
- L2C = 0: No secondary cache
- L2D = 000: L2 cache hit delay (don’t care)
- PLDP = 0: Pipeline depth = 1
- EAV = 1: Drive full address on 60x bus
- ETM = 1: Enable Extended Transfer Mode

- LETM = 1: Enable Local Extended Transfer Mode
- NPQM = 111: Non PowerQUICC II master connected
- EXDD = 0: External Master Delay not disabled
- ISPS = 0: Internal Space Port Size = 64 bits

The resulting register value is BCR=0xA01C0000.

System Protection Control Register (SYPCR)

This register controls the software watchdog. It can be read at any time but can be written only once after system reset. During the first phases of a development, it may be simpler to disable the watchdog by setting SWE to 0 in this register just after reset.

The resulting register value is SYPCR=0xFFFFFFC0.

60x Bus Arbiter Registers (PPC_ACR, PPC_ALRH, and PPC_ALRL)

In the PPC_ACR register, the following fields must be initialized:

- DBGD = 1: Assert –DBG after –TS (needed if bus is parked on the PowerSpan)
- EARB = 0: Internal Arbiter used
- PRKM = 0110: Bus parked on internal PowerPC core

Registers PPC_ALRH and PPC_ALRL define the priorities of the various bus masters. On the 60x bus the recommended priority order is as follows (from the highest to the lowest):

- CPM high priority: highest
- CPM middle priority
- CPM low priority
- External Master (the PowerSpan)
- PowerPC core

The resulting registers values are: PPC_ACR = 0x26, PPC_ALRH = 0x01276345, and PPC_ALRL = 0x89ABCDEF.

SIU Module Configuration Register (SIUMCR)

The SIUMCR register configures various features in the SIU module, among them the configuration of several multifunction pins. Its fields must be set as follows:

- BBD = 0: –ABB and –DBB enabled
- ESE = 1: –GBL/–IRQ1 pin used as –GBL
- PBSE = 0: –PPBS/PGPL4 used as PGPL4
- CDIS = 0: Core is enabled
- DPPC = 00: –IRQ/DP pins used as –IRQ
- L2CPC = 10: L2 cache pins configured as BADDR
- LBPC = 00: Local bus pins used as local bus

- APPC = 00: Address Parity pins used as local bus
- CS10PC = 01: –CS10/–BCTL1 used as –BCTL1
- BCTLC = 01: –BCTL0 used as R/–W and –BCTL1 used as –OE
- MMR = 11: External bus requests initially masked at boot, then
  MMR = 00: No bus request masking once booted
- LPBSE = 0: LBPS/LGPL4 functions as LGPL4

The resulting register value is SIUMCR=0x4205C000.

Bus Transfer Error Registers (TESCR1 and L_TESCR1)

Since there is no parity checking on the 4538, data errors must be disabled (field DMD=1 in registers TESCR1 and L_TESCR1).

Memory Controllers

The PowerQUICC II includes sophisticated memory controller units: a General Purpose Chip-select Machine (GPCM), three User Programmable Machines (UPMs) and two SDRAM control machines. These units are used on the 4538 to control all the external devices, except the PowerSpan, which is directly a 60x bus compatible device.

The memory controller unit to be used is defined bank per bank. Each bank is defined by its Base Register (BRx) and its Option Register (ORx). The memory machine selection is done in the Option register.

Table 2-2. PowerQUICC II Memory Controller Machine Usage

Element Accessed	Bank	Memory Controller	ORx Value	BRx Value
FLASH EEPROM	0	60x bus GPCM	0xFF800881	0xFF800801
60x bus Main memory	1	60x bus SDRAM machine	0xFC002CD0	0x00000041
QuadFalc	2	60x bus UPMA	0xFFFF9104	0xF0080881

SDRAM Controller and SDRAM Device Initialization

For the SDRAM controller, a specific PowerQUICC II register (PSDMR for the 60x SDRAM controller and LSDMR for the local SDRAM controller) is used to configure operations pertaining to the SDRAM. This register includes several configuration fields and one Operation field (OP). This Operation field must be used to generate all the special accesses needed to initialize the SDRAM, such as the precharges, the refreshes, and the SDRAM internal Mode register write. This will be useful for generating the complete SDRAM initialization sequence.

To generate a special access, one must first set the OP field in the xSDMR register, and then generate a dummy access to the SDRAM memory.

The sequence for SDRAM device initialization is as follows:

- Precharge all banks (OP=101)

- Refresh the SDRAM eight times (OP=001)
- Write the SDRAM Mode register (OP=011). For the main SDRAM placed on the 60x bus, the row/column address multiplexing is done externally, so the mode register value must be coded in the column address of the dummy access following the PSDMR programming.
- Reset the xDMR register OP field for normal operation (OP=000).

The refresh periods for the SDRAM devices are defined by one common Memory Refresh Timer Prescaler Register (MPTPR) and by two individual SDRAM Refresh Timer Registers (PSTR for the 60x bus and LSTR for the local bus).

On the 4538, no CPM local memory is present.

GPCM Controller Initialization

The initialization of a GPCM controller is done entirely in the bank Option Register (ORx).

On the 4538, the Flash EEPROM is controlled in bank 0 by a GPCM.

UPM Controller Programming

User Programmable Machine A (UPMA) is used to control accesses to the QuadFALC.

MPC603e Core Initialization

For full description of the MPC603e registers, read Motorola documents: *MPC603e a EC606e RISC Microprocessors User's Manual* (ref MPC603EUM/AD) and *PowerPC Microprocessor Family: The Programmer's Reference Guide* (ref MPRPPCPRG–01).

MMU Initialization

The 4538 local memory mapping is organized in such a way that the Block Address Translation (BAT) mechanism can be used rather than the more complicated Segments and Translation Look-aside Buffers (TLB) mechanism.

In the Boot Firmware, the MMU is initialized using the BAT mechanism. The cachable areas are defined in the BAT blocks. Once the IBATx and DBATx special purpose registers initialized, Address Translation is enabled for instruction and data in the Machine State Register (MSR).

Cache Initialization

The data and instruction caches are automatically invalidated after a power-up or after a hard reset, but not after a soft reset. The content of the instruction and data caches are easily invalidated, using the Instruction Cache FLASH Invalidate (ICFI) and the Data Cache FLASH Invalidate (DCFI) control bits in the HID0 register. Each bit must be set and cleared in two consecutive moves to SPR (`mtspr`) operations to the HID0 register.

The instruction and data caches are enabled through bits ICE and DCE of register HID0 respectively. The setting of ICE bit must be preceded by an **`isync`** instruction. The setting of DCE bit must be preceded by a **`sync`** instruction.

Communication Processor Module Initialization

I/O Port Initialization

The CPM I/O ports have to be configured according to their usage (see *Communication Processor Module (CPM) I/O Ports* on page 8). In the Interphase boot firmware, this is done during the early phase of the boot (in `startup.asm`).

Each CPM port is set by four registers in the Internal Register Area: PDIRx, PPARx, PODRx, and PDATx.

Table 2-3. CPM Port Register initialization Values

Register	Address	Init. Value	Comment
PPARA	0xFF010D04	0x00000000	
PSORA	0xFF010D08	0x00000000	
PDATA	0xFF010D10	0x81000000	SWMODE_N=1, COMCLK_N=1
PDIRA	0xFF010D00	0xFD7FFFFF	
PPARB	0xFF010D24	0x00000000	
PSORB	0xFF010D28	0x00000000	
PDATB	0xFF010D30	0x00000000	
PDIRB	0xFF010D20	0x00003449	
PPARC	0xFF010D44	0x00000410	
PSORC	0xFF010D48	0x00000000	
PDATC	0xFF010D50	0x00000000	QuadFalc, Ethernet reset deactivated
PDIRC	0xFF010D40	0xFFFF0FCA	
PPARD	0xFF010D64	0x00000000	
PSORD	0xFF010D68	0x00000000	
PDATD	0xFF010D70	0x00036000	LEDs off
PDIRD	0xFF010D60	0x0F17F007	

CPM RCCR Reset

At boot, it is important to reset the RISC Controller Configuration Register (RCCR) in order to disable any previously loaded CPM microcode and start with the known default CPM microcode.

Programming the Peripherals 3

Overview

This chapter provides information specific to the 4538 board for peripheral programming. Its initial purpose is not to detail how to program the peripherals themselves, for which the developers should refer to the manufacturers data sheets. However, for tricky peripherals, such as T1/E1/J1 framers, some important register programming is detailed. For more details, refer to the 4538 Boot Firmware sources provided with the CD-ROM and referenced (in *italics*) in this chapter. See also the *4538 Built-In Self Test and Monitor Manual*

PowerQUICC II CPM Initialization

The different functions on the CPM are used as follows:

- MCC1 connected to SI1, using TSA1 (128 time slots)
- MCC2 connected to SI2, using TSA2 (128 time slots)
- FCC3 connected to MII interface for Fast Ethernet
- SMC1 used for TTY interface

Serial Interfaces and Time Slot Assigner Initialization

In the CPM, the Time-Slot Assigners (TSAs) are parts of the Serial Interfaces (SIs).

Most TSA programming is done in two 256x16bits SIx RAMs per SI: one for receive and one for transmit. These SIx RAMs are in the PowerQUICC II internal registers area, they are not a part of the PowerQUICC II internal dual-port RAM. The programming of each entry in the SIx RAM determines the routing of a group of serial bits.
See Boot Firmware sources: tst\c\pqtdm.c - Functions vPQTDM_SI_Init_PQII (Disable all TDM and initialize clock route that defines connection of SIx to the clock sources) *and vPQTDM_SI_Init_PQII_PT* (Initialize SIx for pass-through mode test), *vPQTDM_SI_Init_PQII_SW* (Initialize SIx for switched mode test), *vPQTDM_SI_Init_PQII_MUL* (Initialize SIx for multiplexed mode test), *vPQTDM_SI_Init_PQII_IND* (Initialize SIx for independent mode test).

TDM Busses in Multiplexed Direct Mode and in Switched Mode

According to the TDM busses configuration (see *TDM Bus Configurations* on page 34), the SI1AMR register must be set as follows:

- Reserved = 0: This bit should be cleared.
- SADx = 000: Starting bank address for the RAM of TDMa. 000 for first bank, first 32 entries.
- SDMx = 00: SI Diagnostic Mode for TDMa. 00 means normal operation.

- RFSDx = 01: Receive frame sync delay for TDMa. 01 for 1 clock delay.
- DSCx = 0: Double speed clock for TDMa. 0 means the channel clock rate is equal to the data clock.
- CTRx = 1: Common receive and transmit pin clocks for TDMa. 1 means Rx and Tx clocks are common.
- SLx = 1: Sync level for TDMa. 1 means sync active level is 0.
- CEx = 1: Clock edge for TDMa. When DSCx = 1, data sent on the falling edge and received on the rising edge.
- FEx = 0: Frame Sync edge for TDMa. 0 for falling edge.
- GMx = 0: Grant mode for TDMa. 0 for grant mode not used.
- TFSDx = 01: Transmit frame sync delay for TDMa. 01 for 1 clock delay.

Final Result of SI1AMR register is 0x0171.

TDM Busses in Independent Direct Mode

According to the TDM busses configuration (see *TDM Bus Configurations* on page 34), the SI1AMR, SI1BMR, SI2AMR and SI2BMR registers must be set as follows:

- Reserved = 0: This bit should be cleared.
- SADx = 000: Starting bank address for the RAM of TDMAx. 000 for first bank, first 32 entries.
- SADx = 010: Starting bank address for the RAM of TDMBx. 010 for second bank, first 32 entries.
- SDMx = 00: SI Diagnostic Mode for TDMx. 00 means normal operation.
- RFSDx = 00: Receive frame sync delay for TDMx. 00 for no clock delay.
- DSCx = 0: Double speed clock for TDMx. 0 means the channel clock rate is equal to the data clock.
- CTRx = 1: Common receive and transmit pin clocks for TDMx. 1 means Rx and Tx clocks are common.
- SLx = 1: Sync level for TDMx. 1 means sync active level is 0.
- CEx = 1: Clock edge for TDMx. When DSCx = 0, data sent on rising edge and received on falling edge.
- FEx = 0: Frame Sync edge for TDMx. 0 for falling edge.
- GMx = 0: Grant mode for TDMx. 0 for grant mode not used.
- TFSDx = 00: Transmit frame sync delay for TDMx. 00 for no clock delay.

Final Result of SI1AMR and SI2AMR registers is 0x0070.

Final Result of SI1BMR and SI2BMR registers is 0x2070.

TDM Busses in Pass-Through Mode

According to the TDM busses configuration (see *TDM Bus Configurations* on page 34), the SIxAMR, SixBMR, SIxCMR, and SIxDMR registers must be set as follows (x=1 for line 1 to 2 and line 2 to 1, x=2 for line 3 to 4 and line 4 to 3):

SIxCMR and SIxDMR

- Reserved = 0: This bit should be cleared.
- SADx = 000: Starting bank address for the RAM of TDMs. 010 for second bank, first 32 entries.
- SDMx = 01: SI Diagnostic Mode for TDMs. 01 means automatic echo. In this mode, the TDM transmitter automatically retransmits the TDM received data.
- RFSDx = 01: Receive frame sync delay for TDMs. 01 for 1 clock delay.
- DSCx = 0: Double speed clock for TDMs. 0 means the channel clock rate is equal to the data clock.
- CTRx = 1: Common receive and transmit pin clocks for TDMs. 1 means Rx and Tx clocks are common.
- SLx = 1: Sync level for TDMs. 1 means sync active level is 0.
- CEx = 0: Clock edge for TDMs. When DSCx = 0, data sent on the rising edge and received on the falling edge.
- FEx = 0: Frame Sync edge for TDMs. 0 for falling edge.
- GMx = 0: Grant mode for TDMs. 0 for grant mode not used.
- TFSDx = 01: Transmit frame sync delay for TDMs. 01 for 1 clock delay.

Final Result of SIxCMR and SIxDMR registers is 0x0561.

SIxAMR (line 1 to 2 and line 3 to 4) or SIxBMR (line 2 to 1 and line 4 to 3):

- Reserved = 0: This bit should be cleared.
- SADx = 000: Starting bank address for the RAM of TDM. 000 for first bank, first 32 entries.
- SDMx = 00: SI Diagnostic Mode for TDM. 00 means normal operation.
- RFSDx = 01: Receive frame sync delay for TDM. 01 for 1 clock delay.
- DSCx = 0: Double speed clock for TDM. 0 means the channel clock rate is equal to the data clock.
- CTRx = 1: Common receive and transmit pin clocks for TDM. 1 means Rx and Tx clocks are common.
- SLx = 1: Sync level for TDM. 1 means sync active level is 0.
- CEx = 0: Clock edge for TDM. When DSCx = 0, data sent on the rising edge and received on the falling edge.
- FEx = 1: Frame Sync edge for TDMa. 1 for rising edge.
- GMx = 0: Grant mode for TDM. 0 for grant mode not used.
- TFSDx = 01: Transmit frame sync delay for TDM. 01 for 1 clock delay.

Final Result of SIxAMR (line 1 to 2 and line 3 to 4) and SIxBMR (line 2 to 1 and line 4 to 3) registers is 0x0169.

NOTE

When a TDM is not used, it is not necessary to initialize the corresponding SIxMR register.

By setting CMXSI1CR to 0x30, CLK1 is assigned as input clock to TDMa1, CLK3 is assigned as input clock to TDMb1, CLK13 is assigned as input clock to TDMc1, and CLK15 is assigned as input clock to TDMa1
By setting CMXSI2CR to 0x00, CLK1 is assigned as input clock to TDMd2, CLK3 is assigned as input clock to TDMc2, CLK13 is assigned as input clock to TDMa2, and CLK15 is assigned as input clock to TDMb2.
See Boot Firmware sources: tst\c\pqtdm.c - Function vPQTDM_SI_Init_PQII.

PC(31) must be configured as CLK1 input, PC(29) as CLK3 input, PC(19) as CLK13 input, and PC(17) as CLK15 input.
See Boot Firmware sources: sys\h\4538.h (search CLK1, CLK3, CLK13 and CLK15).

Other TDMx signals also have to be configured on the parallel ports.
See Boot Firmware sources: sys\h\4538.h.
Assign TDMA1_L1RSYNC, TDMA1_L1RXD, and TDMA1_L1TXD to PA(6), PA(8), and PA(9) respectively.
Assign TDMB1_L1RSYNC, TDMB1_L1RXD, and TDMB1_L1TXD to PD(10), PD(12), and PD(13) respectively.
Assign TDMC1_L1RSYNC, TDMC1_L1RXD, and TDMC1_L1TXD to PD(26), PD(27), and PD(28) respectively.
Assign TDMD1_L1RSYNC, TDMD1_L1RXD, and TDMD1_L1TXD to PD(23), PD(24), and PD(25) respectively.

Assign TDMA2_L1RSYNC, TDMA2_L1RXD, and TDMA2_L1TXD to PD(20), PD(21), and PD(22) respectively.
Assign TDMB2_L1RSYNC, TDMB2_L1RXD, and TDMB2_L1TXD to PB(29), PB(30), and PB(31) respectively.
Assign TDMC2_L1RSYNC, TDMC2_L1RXD, and TDMC2_L1TXD to PB(24), PB(26), and PB(27) respectively.
Assign TDMD2_L1RSYNC, TDMD2_L1RXD, and TDMD2_L1TXD to PB(20), PB(22), and PB(23) respectively.

NOTE

TDMbx, TDMCx, and TDMdx parallel port pins must be configured as general purpose output pins.
See Boot Firmware sources: sys\h\4538.h.

Clocks and Baud-Rate Generators

Introduction

The CPM contains eight independent, identical, Baud-Rate Generators (BRGs) that can be used with the FCCs, SCCs, and SMCs. The clocks produced by the BRGs are sent to the bank-of-clocks selection logic, where they can be routed to the controllers. In addition, the output of a BRG can be routed to a pin to be used externally.

BRGCLK

The BRGCLK is an internal signal generated in the MPC8260 clock synthesizer specifically for the BRGs, the SPI, and the I^2C internal BRG. BRGCLK is itself sourced from VCO_OUT (twice the CPM clock) which is at 266.144 MHz. The DFBRG field of SCCR must be programmed to 01, so that BRGCLK equals VCO_OUT/16 (= 16.384 MHz). For more information on SCCR and DFBRG fields, see the MPC8260 PowerQUICC II Users Manual.
See Boot Firmware sources: app\asm\startup.asm.

BRG7 – TTY Baud-Rate Generator

The TTY interface is controlled by SMC1.

SMC1 baud-rate generator is BRG7.

Configure the CMXSMR register as follows:

- SMC1 = 0: SMC1 is not connected to TSA.
- Reserved = 0: This bit should be cleared.
- SMC1CS = 01: SMC1 transmit and receive clocks are BRG7.
- SMC2 = 0: SMC2 is not connected to TSA (don't care).
- Reserved = 0: This bit should be cleared.
- SMC2CS = 00: SMC2 transmit and receive clocks are BRG2 (don't care).

Final Result of CMXSMR register is 0x10. For more information on CMXSMR fields, see the MPC8260 PowerQUICC II Users Manual.

The DIV16 field of BRGC7 register must be set to 0, so the first BRG7 divider will divided the received BRGCLK clock by 1 and will use the 16.384 MHz clock.

To provide the proper baud-rate value (2400, 4800, 9600,... baud), the SMC1 clock source must be 16 times the rate of the line (see BRGC7 register).
See Boot Firmware sources: app\c\montty.c - Function gwMonTTYOpen.

MCC Initialization

In Multiplexed Direct Mode and in Switched Mode, only MCCs TDMa1 is connected to the framers. TDMa2, TDMbx, TDMcx, and TDMdx are not used. TDMa1 can transport up to 128 MCC channels. This must be configured in the MCCF1 register.

MCCF1 register initialization:

- Group 1 = 00: Group 1 (MCC channels 0-31) is used by TDMa1
- Group 2 = 00: Group 2 (MCC channels 32-63) is used by TDMa1
- Group 3 = 00: Group 3 (MCC channels 64-95) is used by TDMa1
- Group 4 = 00: Group 4 (MCC channels 96-127) is used by TDMa1

Final Result of MCCF1 register is 0x00.

MCCF2 register initialization:

- Group 1 = 00: Group 1 (MCC channels 128-159) is used by TDMa2
- Group 2 = 00: Group 2 (MCC channels 160-191) is used by TDMa2
- Group 3 = 00: Group 3 (MCC channels 192-223) is used by TDMa2
- Group 4 = 00: Group 4 (MCC channels 224-255) is used by TDMa2

Final Result of MCCF2 register is 0x00 (don’t care).

In Independent Direct Mode and in Pass-Through Mode, only MCCs TDMax and TDMbx are connected to the framers. TDMcx and TDMdx are not used.

MCCF1 register initialization:

- Group 1 = 00: Group 1 (MCC channels 0-31) is used by TDMa1
- Group 2 = 00: Group 2 (MCC channels 32-63) is used by TDMa1
- Group 3 = 01: Group 3 (MCC channels 64-95) is used by TDMb1
- Group 4 = 01: Group 4 (MCC channels 96-127) is used by TDMb1

Final Result of MCCF1 register is 0x05.

MCCF2 register initialization:

- Group 1 = 00: Group 1 (MCC channels 128-159) is used by TDMa2
- Group 2 = 00: Group 2 (MCC channels 160-191) is used by TDMa2
- Group 3 = 01: Group 3 (MCC channels 192-223) is used by TDMb2
- Group 4 = 01: Group 4 (MCC channels 224-255) is used by TDMb2

Final Result of MCCF2 register is 0x05.

For details on MCC Initialization, *See Boot Firmware sources: tst\c\pqtdm.c Function vPQTDM_MCC_Init_PQII.*

NOTE

The MCCs must be initialized before connecting to them in the SIRAM, otherwise unpredictable errors, such as undue underruns will occur.

T1/E1/J1 Framer Initialization

Introduction

This section details the QuadFALC register initialization, assuming that for non-specified registers, the initialization is the default value (which is generally 0x00). 4538 Boot Firmware sources provides routines to initialize the framers in T1/J1 or E1 mode. Developers should to refer to them.
See Boot Firmware sources: tst\c\qfalc.c)

NOTE

At the end of a QuadFALC port configuration register initialization, it is recommended that you reset the transmitter and receiver by setting XRES and RRES bits in CMDR register.
See Boot Firmware sources: tst\c\qfalc.c - Function vFalcWriteCMDR

Master Clock Initialization

The Master Clock provided on the MCLK pin of the QuadFALC devices is at 12.5 MHz.
See Boot Firmware sources: tst\c\qfalc.c - Functions gvQFalcSetPortSyncSrc and gvQFalcSetPortSyncSrcPT.

Table 3-1. GCM Register Programming

Register	MCLK at 12.5 MHz
GCM1	0x2B
GCM2	0x5D
GCM3	0xAC
GCM4	0x89
GCM5	0x07
GCM6	0x15

TDM Busses General Structure

TDM busses general structure allows four configurations. These different modes are selected by programming the SWMODE_N (PA7) and COMCLK_N (PA0) signals, the TDM ports and the QuadFALC registers.

Multiplexed Direct Mode

In multiplex direct mode, the four framers have the same rhythm. SWMODE_N = 1 and COMCLK_N = 1.

System Interface

QuadFALC is connected to the CPM through an 8 MHz stream. This stream is the concatenation of four 2 MHz streams, corresponding to the four T1/E1/J1 lines. These four streams are mapped into this 8 MHz stream in an interleaved manner.

This interleaved organization is extended to all the 8 MHz streams.

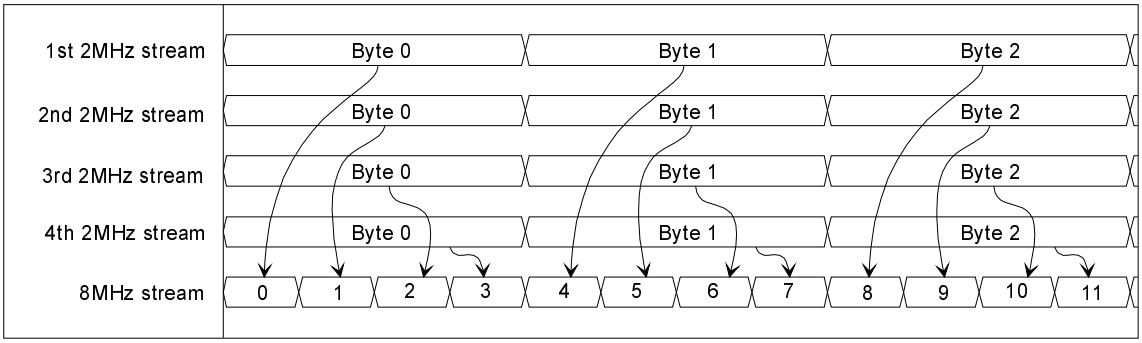


Figure 3-1. Mapping of Four 2 MHz Streams into an 8 MHz Stream

See Boot Firmware sources: tst\c\qfalc.c - Functions gvQFalcInitT1, gvQFalcInitJ1 and *gvQFalcInitE1*.

On the QuadFALC, the system multiplex mode must be enabled (GPC1.SMM = 1) with byte interleaved format (SIC1.BIM=0), clocking rate at 8.192 MHz (SIC1.SCC1/0=10) and data rate at 8.192 MBit/s (SIC1.SDD1=1, FMR1.SDD0=0). Time-slot offset programming was obtained by actual practice: XC0 = 0x00, XC1 = 0x03, RC0 = 0x00, RC1 = 0x03. The receive buffer size must be set to two frames (SIC1.RBS1/0 = 00). The transmit buffer size must be set to two frames (SIC1.XBS1/0 = 10). SIC3.RESX and SIC3.RESR must be set to 0 (Synchronous Pulse Transmit (–SYPX) and Synchronous Pulse Receive (–SYPR) are latched on first clock (8.192 MHz) rising edge). –SYPX and –SYPR inputs are mapped to XPA1 and RPA1 pins respectively by setting the PC1 register to 0. SCLKX_1 and SCLKR_1 must be configured as inputs by setting PC5.CSXP and PC5.CSRP bits to 0. All these initializations must be performed on each channel.

The multiplexed data stream is internally logically ored. Therefore the selection of the active channel phase has to be configured differently for each single channel (1–4). Programming is done with SIC2.SICS2...0 bit as shown in Table 3-2.

NOTE

For T1/J1 applications, the mapping of the receive 24 line time slots over the 32 available on the system interface is configurable with FMR1.CTM bit. In 4538 Boot firmware, the choice is to select 'Channel translation mode 1', by setting FRM1.CTM bit to 1: on reception, the 24 line time slots are contiguously mapped before they are interleaved on the system bus. The same mapping occurs on transmission.

Table 3-2. Channel Phase Programming in Multiplexed System Data Streams

Channel	SIC2.SICS2...0
1	000
2	001
3	010
4	011

RCLK1 Configuration as TDM Bus Clock

See Boot Firmware sources: tst\c\qfalc.c - Function gvQFalcSetPortSyncSrc.

RCLK1 signal of QuadFALC is recovered from the line and dejittered by DCO-R. It must be configured as an active output (PC5.CRP = 1). RCLK2, RCLK3, and RCLK4 shall be configured as inputs (PC5.CRP = 0).

RCLK1 is one of the four channels' internally generated receive route clocks (RCLK) (a channel is a FALC within a QuadFALC) of a QuadFALC: the channel selection is set with GPC1.R1S1 and GPC1.R1S0 bits – when using RCLK1 for synchronizing the TDM SIxRAM, an active channel should be selected. On each channel, program CMR1.RS1=1 and CMR1.RS0=1: the advantage would be to have RCLK1 at 8.192 MHz whatever the source's channel mode is (T1/J1 or E1), the disadvantage is that in case of an LOS (Loss Of Signal) on the source channel, RCLK1 does not go to a continuous level, but is the free running frequency of DCO-R. Since DCO-R is used, program CMR1.DRSS1 and CMR.DRSS0 bits as shown in Table 3-3 to select the reference source for the DCO-R circuit.

Table 3-3. QuadFALC RCLK Reference Source for DCO-R

Channel	CMR1.DRSS1	CMR1.DRSS0
1	0	0
2	0	1
3	1	0
4	1	1

SEC/FSC Configuration

The SEC/FSC signal of the QuadFALC is connected to CPM and is used for the TDM frame synchronization clock (8 KHz synchronization pulse generated by one of the four DCO-Rs). It must be configured as an FSC output by setting GPC1.CSFP1 to 1. Bit GPC1.CSFP0 allows selecting the active level (low or high). When using the pairing feature, FSC source must match an active channel as for RCLK1: the source is selected with GPC1.FSS1 and GPC1.FSS0 bits.
See Boot Firmware sources: TST\C\QFALC.C Function gvQFalcSetPortSyncSrc.

Independent Direct Mode

In independent direct mode, the four framers have their own rhythm. SWMODE_N = 1 and COMCLK_N = 1.

System Interface

QuadFALC is connected to the CPM through four 2 MHz stream, corresponding to the four T1/E1/J1 lines.
See Boot Firmware sources: tst\c\qfalc.c - Functions gvQFalcInitT1, gvQFalcInitJ1 and *gvQFalcInitE1*.

The QuadFALC system multiplex mode must be disabled (GPC1.SMM = 0) with byte interleaved format (SIC1.BIM=0), clocking rate at 2.048 MHz (SIC1.SCC1/0=00) and data rate at 2.048 MBit/s (SIC1.SSD1=0, FMR1.SSD0=0). Time-slot offset programming was obtained by actual practice: XC0 = 0x00, XC1 = 0x03, RC0 = 0x00, RC1 = 0x03. The receive buffer size must be set to two frames (SIC1.RBS1/0 = 00). The transmit buffer size must be set to two frames (SIC1.XBS1/0 = 10). SIC3.RESX and SIC3.RESR must be set to 0 (Synchronous Pulse Transmit (–SYPX) and Synchronous Pulse Receive (–SYPR) are latched on first clock rising edge).
On the first channel, –SYPX (CMR2.IRSP = 0) and –SYPR (CMR2.IXSP = 0) inputs are mapped to XPA1 and RPA1 pins respectively by setting the PC1 register to 0.
SCLKX_1 (used for the transmit system clock CMR2.IXSC = 0) and SCLKR_1 (used for the receive system clock CMR2.IRSC = 0) must be configured as inputs by setting PC5.CSXP and PC5.CSRP bits to 0.
On the other channel, the receive and transmit frame synchronous pulse are internally generated (CMR2.IRSP = 1 and CMR2.IXSP = 1).
SCLKR_x is a 2.048 MHz dejittered receive system clock output (PC5.CSRP = 1) generated by the DCO-R circuit (CMR2.IXSC = 1). The transmit system clock input (PC5.CSXP = 0), SCLKX_x is sourced by the internal receive system clock (CMR2.IRSC = 1).

RCLK1 Configuration as TDM Bus Clock

See Boot Firmware sources: tst\c\qfalc.c - Function gvQFalcSetPortSyncSrc.

RCLK1 signal of QuadFALC is recovered from the line and dejittered by DCO-R. It must be configured as an active output (PC5.CRP = 1). RCLK2, RCLK3, and RCLK4 shall be configured as inputs (PC5.CRP = 0).

RCLK1 is one of the four channels' internally-generated receive route clocks (RCLK) of a QuadFALC: the channel selection is set with GPC1.R1S1 and GPC1.R1S0 bits – when using RCLK1 for synchronizing the TDM SIxRAM, an active channel should be selected. On each channel, program CMR1.RS1=1 and CMR1.RS0=0: the advantage would be to have RCLK1 at 2.048 MHz whatever the source's channel mode is (T1/J1 or E1), the disadvantage is that in case of an LOS (Loss Of Signal) on the source channel, RCLK1 does not go to a continuous level, but is the free running frequency of DCO-R. Since DCO-R is used, program CMR1.DRSS1 and CMR.DRSS0 bits as shown in Table 3-3 to select the reference source for the DCO-R circuit

SEC/FSC Configuration

The SEC/FSC signal of the QuadFALC is connected to CPM and is used for the TDM frame synchronization clock (8 KHz synchronization pulse generated by one of the four DCO-Rs). It must be configured as an FSC output by setting GPC1.CSFP1 to 1. Bit GPC1.CSFP0 allows selecting the active level (low or high). When using the pairing feature, FSC source must match an active channel as for RCLK1: the source is selected with GPC1.FSS1 and GPC1.FSS0 bits.
See Boot Firmware sources: TST\C\QFALC.C Function gvQFalcSetPortSyncSrc.

Switched Mode

In switched direct mode, the four framers have the same rhythm. SWMODE_N = 0 and COMCLK_N = 1.

System Interface

QuadFALC multiplexed bus is connected to the first TDM bus on P4. The second TDM bus on P4 is connected to the MPC8260. TDM busses clock and frame synchronization signals are provided by P4.

On QuadFALC, the system multiplex mode must be enabled (GPC1.SMM = 1) with byte interleaved format (SIC1.BIM=0), clocking rate at 8.192 MHz (SIC1.SCC1/0=10) and data rate at 8.192 MBit/s (SIC1.SDD1=1, FMR1.SDD0=0). Time-slot offset programming was obtained by actual practice: XC0 = 0x00, XC1 = 0x03, RC0 = 0x00, RC1 = 0x03. The receive buffer size must be set to two frames (SIC1.RBS1/0 = 00). The transmit buffer size must be set to two frames (SIC1.XBS1/0 = 10). SIC3.RESX and SIC3.RESR must be set to 0 (Synchronous Pulse Transmit (–SYPX) and Synchronous Pulse Receive (–SYPR) are latched on first clock (8.192 MHz) rising edge). –SYPX and –SYPR inputs are mapped to XPA1 and RPA1 pins respectively by setting the PC1 register to 0. SCLKX_1 and SCLKR_1 must be configured as inputs by setting PC5.CSXP and PC5.CSRP bits to 0. All these initializations must be performed on each channel.

The multiplexed data stream is internally logically ored. Therefore the selection of the active channel phase has to be configured differently for each single channel (1–4). Programming is done with SIC2.SICS2...0 bit as shown in Table 3-2.

See Boot Firmware sources: tst\c\qfalc.c - Functions gvQFalcInitT1, gvQFalcInitJ1 and *gvQFalcInitE1.*

RCLK1 Configuration as TDM bus clock

8 KHz synchronization pulse generated by the internal DCO1-R circuit, synchronized to the lines and provided to P4.

SEC/FSC Configuration

Dejittered clock generated by the internal DCO1-R circuit, synchronized to the lines and provided to P4.

Pass-Through Mode

In multiplex direct mode, the four framers have the same rhythm. SWMODE_N = 1 and COMCLK_N = 0.
Pass-Through mode is possible from framer 1 to framer 2, framer 3 to framer 4 and vice versa.

System Interface

See Boot Firmware sources: tst\c\qfalc.c - Function gvQFalcInitE1PT.

The QuadFALC system multiplex mode must be disabled (GPC1.SMM = 0) with byte interleaved format (SIC1.BIM=0), clocking rate at 2.048 MHz (SIC1.SCC1/0=00) and data rate at 2.048 MBit/s (SIC1.SSD1=0, FMR1.SSD0=0). Time-slot offset programming was obtained by actual practice: XC0 = 0x00, XC1 = 0x04, RC0 = 0x00, RC1 = 0x04. The receive buffer size must be set to two frames (SIC1.RBS1/0 = 00). The transmit buffer size must be set to two frames (SIC1.XBS1/0 = 10). SIC3.RESX and SIC3.RESR must be set to 0 (Synchronous Pulse Transmit (–SYPX) and Synchronous Pulse Receive (–SYPR) are latched on first clock rising edge).
–SYPX (CMR2.IRSP = 0) and –SYPR (CMR2.IXSP = 0) inputs are mapped to XPAx and RPAx pins respectively by setting the PC1 register to 0.
SCLKX_x (used for the transmit system clock CMR2.IXSC = 0) and SCLKR_x (used for the receive system clock CMR2.IRSC = 0) must be configured as inputs by setting PC5.CSXP and PC5.CSRP bits to 0.

RCLK1 Configuration as TDM Bus Clock

See Boot Firmware sources: tst\c\qfalc.c - Function gvQFalcSetPortSyncSrcPT.

RCLK1 signal of QuadFALC is recovered from the line and dejittered by DCO-R. It must be configured as an active output (PC5.CRP = 1). Though RCLK2, RCLK3, and RCLK4 are not connected and shall be configured as inputs (PC5.CRP = 0).

RCLK1 is one of the four channels' internally generated receive route clocks (RCLK) of a QuadFALC: the channel selection is set with GPC1.R1S1 and GPC1.R1S0 bits – when using RCLK1 for synchronizing the TDM SIxRAM, an active channel should be selected. On each channel, program CMR1.RS1=1 and CMR1.RS0=0: the advantage would be to have RCLK1 at 2.048 MHz whatever the source's channel mode is (T1/J1 or E1), the disadvantage is that in case of an LOS (Loss Of Signal) on the source channel, RCLK1 does

not go to a continuous level, but is the free running frequency of DCO-R. Since DCO-R is used, program CMR1.DRSS1 and CMR.DRSS0 bits as shown in Table 3-3 to select the reference source for the DCO-R circuit

SEC/FSC Configuration

The SEC/FSC signal of the QuadFALC is connected to CPM and is used for the TDM frame synchronization clock (8 KHz synchronization pulse generated by one of the four DCO-R). It must be configured as an FSC output by setting GPC1.CSFP1 to 1. Bit GPC1.CSFP0 allows selecting the active level (low or high). When using the pairing feature, FSC source must match an active channel as for RCLK1: the source is selected with GPC1.FSS1 and GPC1.FSS0 bits.
See Boot Firmware sources: TST\C\QFALC.C Function gvQFalcSetPortSyncSrc.

Framing and Line Coding Initialization

Common Initialization

Table 3-4. Common T1/E1/E1-CRC4 Initialization

Register Bit	Value	Comment
LIM1.DRS	0	The ternary interface is selected.
LIM2.SLT1...0	10	Receive slicer threshold = 50%
LIM1.RIL2...0	010	Line interface receive input threshold equals 0.6 V
PCD,PCR	0x0A,0x15	LOS is declared after 176 pulse positions without transitions and 0x16 pulses required within 176 ms to clear an LOS alarm (fulfills G.775).

T1 Specific Initialization

Table 3-5. T1 Specific Initialization

Register Bit	Value	Comment
FMR1.PMOD	1	T1 mode
FMR4..FM1...0	2	24-frame multi-frame format (ESF)
FMR1.CRC	1	CRC6 check/generation enabled
FMR0.RC1...0	11	B8ZS serial line code for the receiver
FMR0.XC1...0	11	B8ZS serial line code for the transmitter
FRM4.AUTO	1	Automatic re-synchronization is enabled

E1/E1-CRC4 Common Initialization

Table 3-6. E1/E1-CRC4 Common Initialization

Register Bit	Value	Comment
FMR1.PMOD	0	E1 mode
FMR0.RC1...0	11	HDB3 serial line code for the receiver.
FMR0.XC1...0	11	HDB3 serial line code for the transmitter.
XSW.XSIS	1	Spare bit for International Use: not used.
XSW.XY0...4	1111	Spare bits for National Use: not used.

E1 Non-CRC4 Specific Initialization

Table 3-7. E1 Non-CRC4 Specific Initialization

Register Bit	Value	Comment
FMR1.XFS	0	Transmit double-frame format.
FMR2.RFS1...0	00	Receive double-frame format.

E1-CRC4 Specific Initialization

Table 3-8. E1-CRC4 Specific Initialization.

Register Bit	Value	Comment
FMR1.XFS	1	Transmit CRC4-multiframe format.
FMR2.RFS1...0	10	Receive CRC4-multiframe format.

Clock Synchronization Initialization

Slave Mode

Table 3-9. Slave Mode Initialization

Register Bit	Value	Comment
LIM0.MAS	0	Slave mode.
LIM2.ELT	1	Enable Loop-Timed. Transmit clock is generated from the clock supplied by MCLK which is synchronized to the extracted receive route clock.
FMR5.XTM	1	Disconnects the control of the transmit system interface from the transmitter.
CMR1.DXSS	0	The DCO-X circuitry synchronizes to the internal reference clock which is sourced by RCLK (assuming LIM1.RL=0 and LIM2.ELT = 1).

Master Mode

Table 3-10. Master Mode Initialization

Register Bit	Value	Comment
LIM0.MAS	1	Master mode.
LIM2.ELT	0	Normal operation – Loop-Timed disabled.
FMR5.XTM (T1) XSW.XTM (E1)	0	–SYSPX defines the frame beginning on the transmit system highway.
IPC.SSYF	0	Reference clock at port SYNC is 2.048 MHz.

Table 3-10. Master Mode Initialization (cont)

Register Bit	Value	Comment
LIM1.DCOC (T1)	1	For T1 only – 2.048 MHz reference clock for the DCO-R circuitry provided on pin SYNC.

Transmit Pulse Shape

For each type of Line Build-Out (LBO), the shape of the transmit pulse must be adjusted through QuadFALC registers LIM0, LIM2, XPM0, XPM1, and XPM2 in order to comply with FCC 68 or ANSI T1.403 (see Table 1-27 on page 31).

Line LED Control

For each T1/E1/J1 line, there is one green LED.

Each green LED is controlled through a QuadFALC pin (RPDi), allowing software or hardware control. Each of these RPDi pins can be configured for one of seven different functions. The selection is done in QuadFALC register PC4 (one for each pin). This gives the ability to control the LED in different operations:

- When PC4 = 0x00, the green LED is OFF
- When PC4 = 0x30, the green LED is ON
- When PC4 = 0x70, the green LED is ON if synchronized to Rx line, OFF otherwise.

The Ethernet Port Initialization

The Ethernet Line Interface Unit (LIU) is a INTEL LXT971A.

The LIU is connected to FCC3 through a Media Independent Interface (MII).
See Boot Firmware: sys\h\4538.h.

The LIU internal registers are initialized through MDC and MDIO Management pins. These pins have to be manually manipulated through PC(25) and PC(26) pins. The LIU PHY address is set to 0 (address pins are cabled to 0V).
See Boot Firmware: eth\c\lxtinit.c.

The TTY Framer Initialization

The TTY port is connected to an SMC1 framer that is used in UART mode. For SMC1 operation, MPC8260 port D pins 8 and 9 have to be configured properly.

See Boot Firmware: sys\h\4538.h.

SMC1 baud-rate generator is BRG7 - see *BRG7 – TTY Baud-Rate Generator* on page 73.

For a simple SMC1 controller example in polling mode:
See Boot Firmware: app\c\montty.c

Accessing the 4538 on the PCI Side

4

PowerSpan Configuration by the PCI Host

Several elements of the PowerSpan are automatically configured at power-up by the hardware, or by the PowerQUICC II. However, some PCI-specific settings have to be done by the PCI host.

PCI Configuration

The card is identified through its Interphase Vendor ID (0x107E) and its PCI device ID (0x9070). Its PCI configuration is set up by the PCI host at its power-on or by the “high availability” operating system if the 4538 has been hot inserted.

Interrupt Pin Configuration

The set up of the PowerSpan Interrupt Map registers is normally done by the PowerQUICC II when it boots, so they should not need to be reconfigured, except if the card has not yet received a valid boot firmware.

PCI-to-Local Window Configuration

When accessing through a PowerSpan PCI-to-local window, this window must have been enabled in the I^2C serial EEPROM, in order to allow the CompactPCI host to detect it at system power-on or after hot insertion of the board, and map it in the PCI space.

The corresponding PowerSpan register PCI Target Image Control Register must also have been initialized with the Image Enable bit set (IMG_EN=1) and the address translation mechanism enabled (TA_EN=1).

Controlling the 4538 Hardware and Software Resets

PowerSpan interrupt pins –INT2 and –INT3 are used as output ports to control the MPC8260 hardware reset signal –HRESET and software reset signal –SRESET respectively. They are conventionally associated with doorbell bits 4 and 5 respectively. The PowerSpan Interrupt Map registers must have been correctly initialized before (see *PowerSpan Configuration by the PCI Host* on page 87).

During a power-up sequence, –HRESET and –SRESET are first activated and then deactivated once the PCI bus reset signal is deactivated. This allows the PowerQUICC II to boot without any host intervention, just after the end of the PCI reset.

For a normal use, the card should be reset by the PCI host (if needed) using only the –SRESET signal. The –HRESET signal is used for special cases, such as FLASH memory re-programming through PCI. Example 4-1 is an example of C code routines to reset and run the board from the PCI side.

Example 4-1. Reset and Run Command Routines

```
#define T_HRESET        0x1000// DB4 controls -INT2 as output for -HRESET
#define T_SRESET        0x2000// DB5 controls -INT3 as output for -SRESET


void ResetCard( void)
{
    RegWrite32( T_IER0, RegRead32( T_IER0) | T_SRESET);
}


void RunCard( void)
{
    RegWrite32( T_ISR0, T_HRESET | T_SRESET);
}
```

Controlling the PCI-to-Local Interrupt

The PowerSpan Interrupt pin –INT0 is used to control the PCI-to-Local interrupt (renamed ATN in the software examples: “Attention to the PowerQUICC II”). It is associated by convention with doorbell register 2. The PowerSpan Interrupt Map registers must have been previously correctly initialized. This interrupt controls the –IRQ1/DP1/–EXT_BG2 input pin of the PowerQUICC II.

Example 4-2 is an example of C code routines to set and reset the PCI-to-Local interrupt and to read the status of this interrupt from the PCI side.

Example 4-2. PCI to Local Interrupt Routines (From the PCI Side)

```
#define T_ATN            0x400    // DB2 controls -INT0 as output for -ATN
void SetAtn( void)
{
    RegWrite32( T_IER0, T_ATN);
}
unsigned char AtnState( void)
{
    return( (unsigned char) ((RegRead32( T_ISR0) & T_ATN)? 1 : 0) );
}
```

Local to PCI Interrupt (–INTA)

The PowerQUICC II can generate an interrupt toward the PCI Host by setting a doorbell bit. Conventionally, doorbell bit 0 has been dedicated to this task, and has been associated with PCI interrupt pin –INTA in the PowerSpan Interrupt Map registers.

Example 4-3 is an example of C code routines to reset the PCI-to-Local interrupt and to read the status of this interrupt from the local side.

Example 4-3. Routines Related to Local-to-PCI Interrupt

```
#define T_INTA 0x100    // DB0 controls -INTA PCI interrupt
void ResetIntA( void)
{
    RegWrite32( T_IER0, RegRead32( T_IER0) | T_INTA);
}
unsigned char PciIntState( void)
{
    return( (unsigned char) ((RegRead32( T_ISR0) & T_INTA)? 1 : 0) );
}
```

Local Space Access From PCI Memory Space

The PowerSpan provides four memory windows from the PCI memory space to the Local memory space. In the 4538 design, the default setting in the PowerSpan serial EEPROM enables two windows. The first one is set with a size of 2 MB and is intended for “operational” exchanges. The second one is set with a size of 512 KB and is intended to be used for “dumps”.

During a PCI host access to local space, the high-order address bits of the local bus must be generated by the PowerSpan (as defined in the PowerSpan P1_TI0_ADDR register), the low-order address bits of the local bus come from the PCI address. This mode is called “Address Translation” in the PowerSpan manual.

NOTE

When accessing through a PowerSpan PCI-to-local window, this window must have been enabled in the I^2C serial EEPROM, in order to allow the PCI host to detect it at system power-on or after hot insertion of the board, and map it in the PCI space. The corresponding PowerSpan register “PCI Target Image Control Register” must also have been initialized with the “Image Enable” bit set (IMG_EN=1) and the address translation mechanism enabled (TA_EN=1).

When the processor is running, the PCI bus has access to all the elements connected to the local bus, except the FLASH boot memory: the main SDRAM memory (the processor's SDRAM memory controller must be initialized), the QuadFALC framers, etc. (the processor must have its chip selects programmed). Local space mapping is the same as when accessed by the processor.

It is not possible to have access to the entire FLASH device when the processor is running, because the FLASH device is an 8-bit data bus device connected to the 64-bit-only local bus of the PowerSpan. Only bytes modulo 8 are reachable.

This problem has been neutralized for the other non-64-bit peripherals, by tying their peripheral address bits 0 to N to the local address bits 3 to N+3 respectively, so that all their registers can be accessed on byte lane 0, at consecutive modulo 8 addresses.

When the processor is in the reset state, its memory controllers and chip-select signals are reset, so nothing can be accessed.

Access to the FLASH EEPROM Through PCI

For FLASH in-situ re-programming by the PCI host, a special FLASH mode provides access. In this mode, the PowerQUICC II is reset and logic generates a FLASH chip-select and overcomes the problem of an 8-bit device connected to a 64-bit-only PowerSpan.

The specific FLASH mode is enabled by one of the PowerSpan interrupt pins (–INT1) used as an output port. When –INT1 is set to 0, the PowerQUICC II is maintained in hard reset state (–HRESET=0), its pins are tri-stated, the 60x bus is parked for the PowerSpan, and the following address bus remap is implemented: the device's low order address bit A (2:0) is driven by the PowerSpan address bit A (24:22). This remap allows full access of the FLASH content through byte lane 0 of the 64-bit 60x bus, provided that some address translation is done by the software.

Example 4-4. Set and Reset FLASH Mode Routine (From PCI Side)

```
#define T_FLASH_EN      0x800    // DB3 controls -INT1 as output for -FLASH_EN
#define T_HRESET        0x1000    // DB4 controls -INT2 as output for -HRESET
#define T_SRESET        0x2000    // DB5 controls -INT3 as output for -SRESET

void SetFLASHMode( void)
{
    RegWrite32( T_IER0, RegRead32( T_IER0) | T_SRESET | T_HRESET); // Reset
    RegWrite32( T_IER0, RegRead32( T_IER0) | T_FLASH_EN); // then FLASH mode
}

void ResetFLASHMode( void)
{
    RegWrite32( T_ISR0, T_FLASH_EN); // Reset FLASH mode
    RegWrite32( T_ISR0, T_HRESET | T_SRESET); // then run
```

```
}
```

Example 4-5. FLASH Read and Write Routines (From PCI Side)

```
unsigned char FLASHRead( unsigned long addr)
{
    unsigned char data, *ptr;

    /*----- Move the Memory window over the FLASH area -----*/
    RegWrite32( T_P1_TI0_ADDR, FLASH.BaseAddr);

    /*----- Put address bit A(2:0) at address bits A(24:22) -----*/
    addr = ((addr & 7)<<22) | (addr & 0x3FFFF8);

    /*----- Do the read -------*/
    ptr  = (unsigned char*)(WinBase+addr);
    data = *ptr;

    /*----- Move back Memory window at 0 -----*/
    RegWrite32( T_P1_TI0_ADDR, 0);
    return data;
}

void FLASHWrite( unsigned long addr, unsigned char data)
{
    unsigned char* ptr;

    /*----- Move the Memory window over the FLASH area -----*/
    RegWrite32( T_P1_TI0_ADDR, FLASH.BaseAddr);

    /*----- Put address bit A(2:0) at address bits A(24:22) -----*/
    addr = ((addr & 7)<<22) | (addr & 0x3FFFF8);

    /*----- Do the write -------*/
    ptr  = (unsigned char*)(WinBase+addr);
    *ptr = data;

    /*----- Move back Memory window at 0 -----*/
    RegWrite32( T_P1_TI0_ADDR, 0);
}
```

FLASH EEPROM Programming Algorithms

The boot memory is a 4Mx8 AMD 29LV033 FLASH device. To reprogram the AMD FLASH device, special programming algorithms are defined by AMD, which combine reads and writes with special address patterns. The algorithm descriptions can be found at the AMD web site. You can also look or start from the source provided in the BDK (file `app\c\amdflash.c`).

Serial EEPROM Connected to the PowerSpan

An I²C serial EEPROM is connected to the PowerSpan. It is used to store certain PowerSpan register initialization values and the PCI Vital Product Data (VPD). Other Interphase-specific data is stored there, and there is still some room for other custom data. See “Serial EEPROM Connected to the PowerSpan” on page 27.

Table 2-1 on page 60 provides the PowerSpan Register initialization values stored in the Serial EEPROM.

The I²C Serial EEPROM can be easily accessed from the PCI side or from the local processor side, by using dedicated PowerSpan Register I2C_CSR.

Example 4-6 is an example of C code read and write routines.

Example 4-6. I²C Serial EEPROM Read and Write Routines (From PCI Side)

```
unsigned char EepromByteRead( unsigned char a)
{
    unsigned long v;
    v = ((unsigned long) a)<<24 | 0xA000;
    while( RegRead32( T_I2C_CSR) & 0x80);  // Wait ACT=0
    RegWrite32( T_I2C_CSR, v);
    while( (v=RegRead32( T_I2C_CSR) & 0x80));  // Wait ACT=0
    if (v&0x40) printf("error");
    v>>=16;
    return (unsigned char) v;
}

void EepromByteWrite( unsigned char a, unsigned char d)
{
    unsigned long v,s;
    v = ((unsigned long) a)<<24 |((unsigned long) d)<<16 | 0xA100;
    while((s=RegRead32( T_I2C_CSR)) & 0x80);  // Wait ACT=0
    if (s&0x40) printf("error");
    RegWrite32( T_I2C_CSR, v);
}
```

In Situ EPLD Programming

Glue logic is implemented in some EPLDs that can be programmed in the field using the PCI interface.

The EPLDs are in a daisy-chain configuration, which enables all of them to be programmed at once. They can be programmed in-situ by the PCI host, using PowerSpan interrupts as I/O pins. A jumper must be placed on board location JP1 to enable the programming (when present, this jumper sets the ISP signal –ISPEN to its active state 0).

These devices are initialized by Interphase and keep their programming during power off. The normal user should not need to reprogram them.

Optimizing the PCI Bus Utilization

The PCI maximum throughput of 266 MB/s is very difficult to reach. The actual throughput can be very disappointing if certain principles are not followed. These principles are:

Avoid the reads. Prefer the writes. Writes can be very efficient, because they are posted in the FIFOs included in the various PCI bridges. A read needs completion of the entire data transfer from its origin to its destination, before being considered as finished. Because of the arbitrations on the various local busses and because of resynchronizations occurring each time there are different bus clocks, a single read can take approximately 1 µs.

Prefer the bursts. During a burst, the duration of the transfers after the first one can be very efficient and last only one PCI cycle. On the 4538, only the PowerSpan DMAs can generate efficient bursts, because they do transfers to incremental addresses.

Prefer DMA transfers. For data transfers between the PCI space and the local 60x memory, the PowerSpan DMAs are more efficient than the local processor. They can use bursts on both the local 60x side and on the PCI side. They use FIFOs to de-couple the PCI bandwidth and the 60x bandwidth occupancies.

Effective Ordering of the PCI Accesses

The PowerSpan includes FIFOs between the PCI bus and the 60x bus in each direction. When a write is done by the PCI host into the local memory, the PowerSpan can acknowledge this write as soon as there is a place in the FIFO, but the effective write into the local memory can be delayed, due to previous writes still waiting in the FIFO, or due to the local 60x bus being used by the processor.

If the PCI Host makes an access to a PowerSpan register, just after this write to the local memory, the effective completion of this register access may occur before the effective write into the local memory. This can lead to unexpected behavior.

The order in which the PCI makes successive writes and reads into the local memory may also not be respected on the local side. Suppose that the host makes several writes followed by one read. Because the FIFO in the write direction may take some time to get emptied on the local side, the effective read on the local side may happen before the last write.

Example: The PCI host sets the DMA buffer descriptors into the local memory, and then it runs the DMA (a write into a PowerSpan register). The DMA starts before the effective completion of the buffer descriptors writes into the local 60x memory, so it loads a bad addresses, a bad byte count, etc., and accomplishes the transfer with this bad data.

In this case, the PCI host must ensure that its latest write into the memory is effectively finished locally, before starting the DMA.

PCI Deadlock Situations

Several deadlock situations can occur on the PCI bus. These situations will statistically rarely occur, but they need to be treated by exception handlers, in order not to lock the system.

Connector Placement

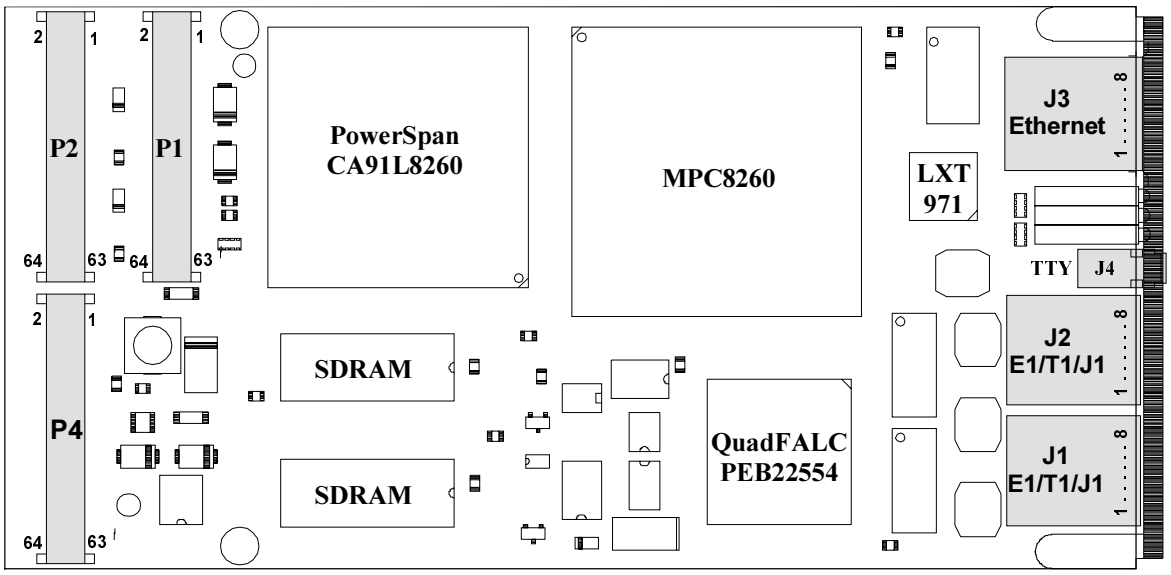


Figure 5-1. Connectors on the Component Side

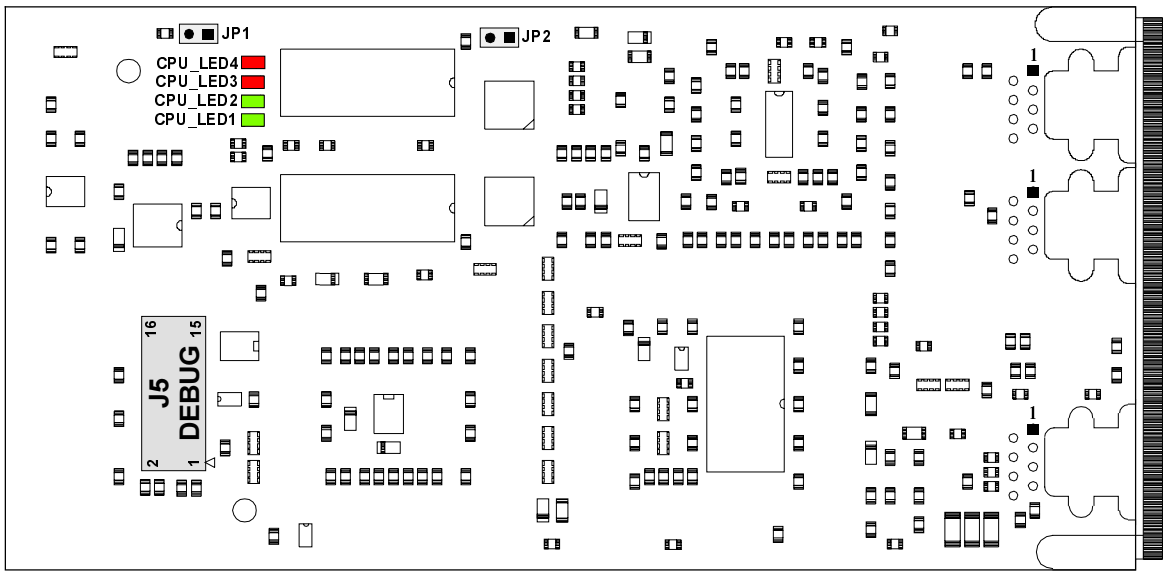


Figure 5-2. Connectors and LEDs on the Solder Side

Front Panel

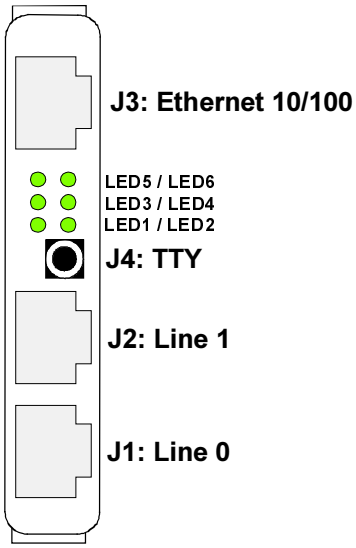


Figure 5-3. Connectors and Leds on front panel

LED Descriptions

CPU_LED1: Board user-programmable green LED controlled by PD(15)

CPU_LED2: Board user-programmable green LED controlled by PD(14)

CPU_LED3: Board user-programmable red LED controlled by PD(18)

CPU_LED4: Board user-programmable red LED controlled by PD(17)

LED1:Synchronization signal provided by the Framer 1 for Line 0

LED2:Synchronization signal provided by the Framer 2 for Line 1

LED3: LXT971 LED driver 1

LED4: LXT971 LED driver 2

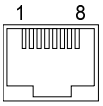
LED5: LXT971 LED driver 3

LED6: User-programmable LED, CPU_LED6 controlled by PD(16)

RJ48 Connectors J1 and J2

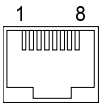
J1 is tied to the first framer and J2 is tied to the second framer.

Table 5-1. RJ48 Connectors J1 and J2

	Signal
1	IN1
2	IN2
3	
4	OUT1
5	OUT2
6	
7	
8	

Ethernet 10/100 RJ45 Connector J3

Table 5-2. Ethernet 10/100 RJ45 Connector

	Signal
1	OUT+
2	OUT-
3	IN+
4	
5	
6	IN-
7	
8	

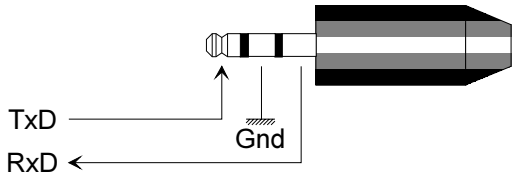
TTY Serial Port J4

A 2.5mm stereo jack connector provides a connection to an asynchronous serial device such as a TTY console. Signals on this connector have EIA-232-D electrical levels (RS232) for direct connection to a console.

Table 5-3. J4 TTY Serial Connector

Pin	Signal
Ring	Ground
Tip	TxD
Sleeve	RxD

Figure 5-4. TTY connector : 2.5mm stereo jack plug



PMC Connectors

PMC Connectors P1 and P2

PMC connectors P1 and P2 support the 32-bit PCI bus as defined by the PMC standard. Signal levels are classified in the “Very Low Voltage Directory” by IEC 950 safety standard.

Table 5-4. PMC Connector P1

No.	Pin Name	Pin Type	Description
1			Not connected.
2			Not connected.
3	GND	Supply	Ground.
4	INTA#	Output open drain	PCI interrupt A: Interrupt from the 4538 to the PCI Host controlled by software by the PowerQUICC II.
5			Not connected.
6			Not connected.
7	BUSMODE1#	Output	Board Presence Indication: indicates the presence and the PCI protocol capability of the board in response to the BUSMODE[4:2]# signals.
8	+5V	Supply	+5 V Supply not used on this board.

Table 5-4. PMC Connector P1 (cont)

No.	Pin Name	Pin Type	Description
9			Not connected.
10			Not connected.
11	GND	Supply	Ground.
12			Not connected.
13	PCI_CLK	Input	PCI Clock: Clock input for the PCI interface. The frequency should be between 25 MHz and 33 MHz.
14	GND	Supply	Ground.
15	GND	Supply	Ground.
16	GNT#	Input	PCI Grant: Input because an external arbiter is used
17	REQ#	Output	PCI Bus Request: Output because an external arbiter is used.
18	+5V	Supply	+5 V Supply not used on this board.
19	VIO	Supply	VIO Supply: 3.3 V or 5.0 V.
20	PAD(31)	Tristate bidirectional	PCI Address/Data.
21	PAD(28)	Tristate bidirectional	PCI Address/Data.
22	PAD(27)	Tristate bidirectional	PCI Address/Data.
23	PAD(25)	Tristate bidirectional	PCI Address/Data.
24	GND	Supply	Ground.
25	GND	Supply	Ground.
26	CBE3#	Tristate bidirectional	PCI Bus Command and Byte Enable.
27	PAD(22)	Tristate bidirectional	PCI Address/Data.
28	PAD(21)	Tristate bidirectional	PCI Address/Data.
29	PAD(19)	Tristate bidirectional	PCI Address/Data.
30	+5V	Supply	+5 V Supply not used on this board.
31	VIO	Supply	VIO Supply: 3.3V or 5.0V.
32	PAD(17)	Tristate bidirectional	PCI Address/Data.
33	FRAME#	Tristate bidirectional	PCI Cycle Frame.
34	GND	Supply	Ground.
35	GND	Supply	Ground.
36	IRDY#	Tristate bidirectional	PCI Initiator Ready.
37	DEVSEL#	Tristate bidirectional	PCI Device Select.
38	+5V	Supply	+5 V Supply not used on this board.
39	GND	Supply	Ground.
40			Not connected.

Table 5-4. PMC Connector P1 (cont)

No.	Pin Name	Pin Type	Description
41			Not connected.
42			Not connected.
43	PAR	Tristate bidirectional	PCI Parity.
44	GND	Supply	Ground.
45	VIO	Supply	VIO Supply: 3.3 V or 5.0 V.
46	PAD(15)	Tristate bidirectional	PCI Address/Data.
47	PAD(12)	Tristate bidirectional	PCI Address/Data.
48	PAD(11)	Tristate bidirectional	PCI Address/Data.
49	PAD(9)	Tristate bidirectional	PCI Address/Data.
50	+5V	Supply	+5 V Supply not used on this board.
51	GND	Supply	Ground.
52	CBE0#	Tristate bidirectional	PCI Bus Command and Byte Enable.
53	PAD(6)	Tristate bidirectional	PCI Address/Data.
54	PAD(5)	Tristate bidirectional	PCI Address/Data.
55	PAD(4)	Tristate bidirectional	PCI Address/Data.
56	GND	Supply	Ground.
57	VIO	Supply	VIO Supply: 3.3 V or 5.0 V.
58	PAD(3)	Tristate bidirectional	PCI Address/Data.
59	PAD(2)	Tristate bidirectional	PCI Address/Data.
60	PAD(1)	Tristate bidirectional	PCI Address/Data.
61	PAD(0)	Tristate bidirectional	PCI Address/Data.
62	+5V	Supply	+5 V Supply not used on this board.
63	GND	Supply	Ground.
64			Not connected.

Table 5-5. PMC Connector P2

No.	Pin Name	Pin Type	Description
1			Not connected.
2			Not connected.
3			Not connected
4	PCITDO	Output	JTAG Test Output: Because the board does not support the IEEE Standard 1149.1 interface, PCITDO and PCITDI pins are hardwired.

Table 5-5. PMC Connector P2 (cont)

No.	Pin Name	Pin Type	Description
5	PCITDI	Input	JTAG Test Input: Because the board does not support the IEEE Standard 1149.1 interface, PCITDO and PCITDI pins are hardwired.
6	GND	Supply	Ground.
7	GND	Supply	Ground.
8			Not connected.
9			Not connected.
10			Not connected.
11	BUSMODE2#	Input	BUSMODE2# Signal: used with BUSMODE3# and BUSMODE4# to determine the presence and the protocol capability of the board. The result is output on BUSMODE1#.
12	+3.3V	Supply	+3.3 V Supply.
13	PCI_RST#	Input	PCI Reset.
14	BUSMODE3#	Input	BUSMODE3# Signal: used with BUSMODE2# and BUSMODE4# to determine the presence and the protocol capability of the board. The result is output on BUSMODE1#.
15	+3.3V	Supply	+3.3 V Supply.
16	BUSMODE4#	Input	BUSMODE4# Signal: used with BUSMODE2# and BUSMODE3# to determine the presence and the protocol capability of the board. The result is output on BUSMODE1#.
17			Not connected.
18	GND	Supply	Ground.
19	PAD(30)	Tristate bidirectional	PCI Address/Data.
20	PAD(29)	Tristate bidirectional	PCI Address/Data.
21	GND	Supply	Ground.
22	PAD(26)	Tristate bidirectional	PCI Address/Data.
23	PAD(24)	Tristate bidirectional	PCI Address/Data.
24	+3.3V	Supply	+3.3 V Supply.
25	IDSEL	Input	PCI Initialization Device Select.
26	PAD(23)	Tristate bidirectional	PCI Address/Data.
27	+3.3V	Supply	+3.3 V Supply.
28	PAD(20)	Tristate bidirectional	PCI Address/Data.
29	PAD(18)	Tristate bidirectional	PCI Address/Data.
30	GND	Supply	Ground.

Table 5-5. PMC Connector P2 (cont)

No.	Pin Name	Pin Type	Description
31	PAD(16)	Tristate bidirectional	PCI Address/Data.
32	CBE2#	Tristate bidirectional	PCI Bus Command and Byte Enable.
33	GND	Supply	Ground.
34			Not connected.
35	TRDY#	Tristate bidirectional	PCI Target Ready.
36	+3.3V	Supply	+3.3 V Supply.
37	GND	Supply	Ground.
38	STOP#	Tristate bidirectional	PCI Stop.
39	PERR#	Tristate bidirectional	PCI Parity Error.
40	GND	Supply	Ground.
41	+3.3V	Supply	+3.3 V Supply.
42	SERR#	Output open drain	PCI System Error.
43	CBE1#	Tristate bidirectional	PCI Bus Command and Byte Enable.
44	GND	Supply	Ground.
45	PAD(14)	Tristate bidirectional	PCI Address/Data.
46	PAD(13)	Tristate bidirectional	PCI Address/Data.
47	GND	Supply	Ground.
48	PAD(10)	Tristate bidirectional	PCI Address/Data.
49	PAD(8)	Tristate bidirectional	PCI Address/Data.
50	+3.3V	Supply	+3.3 V Supply.
51	PAD(7)	Tristate bidirectional	PCI Address/Data.
52			Not connected.
53	+3.3V	Supply	+3.3 V Supply.
54			Not connected.
55			Not connected.
56	GND	Supply	Ground.
57			Not connected.
58			Not connected.
59	GND	Supply	Ground.
60			Not connected.
61			Not connected.
62	+3.3V	Supply	+3.3 V Supply.
63	GND	Supply	Ground.

Table 5-5. PMC Connector P2 (cont)

No.	Pin Name	Pin Type	Description
64			Not connected.

PMC Connector P4

PMC connector P4 supports the four E1/T1 lines and two TDM busses with clocks and synchronization signals. The framers 1, 2, 3 and 4 are respectively tied to the lines 0, 1, 2 and 3. Signal levels are classified in the “Very Low Voltage Directory” by IEC 950 safety standard.

Table 5-6. PMC Connector P4

No.	Pin Name	Pin Type	Description
1	DOH0	Input 10 K Pull up to +3.3 V	Data from External TDM bus 0: in "Switched Mode", 8.192 Mb/s transmit data from P4 to the QuadFALC transmitters.
2	DIH0	Tristate Output	Data to External TDM bus 0: in "Switched Mode", 8.192 Mb/s received data from the QuadFALC and sent on P4. This output is enabled only in "switched mode".
3	DOH1	Input 10 K Pull up to +3.3 V	Data from External TDM bus 1: in "Switched Mode", 8.192 Mb/s data from P4 to the MPC8260 TDMa1 bus.
4	DIH1	Tristate Output	Data to External TDM bus 1: in "Switched Mode", 8.192 Mb/s data from the MPC8260 TDMa1 bus sent on P4. This output is enabled only in "switched mode".
5	FSOH	Input 10 K Pull down	External 8 KHz frame synchronization pulse: provided by P4 to the MPC8260 and the QuadFALC in "switched mode". FSOH can also be used as a reference clock for the QuadFALC SYNC input.
6	FSIH	Tristate Output	8 KHz pulse: generated by the QuadFALC and output to the motherboard. This output is enabled only in "switched mode".
7	CKOH	Input 10 K Pull up to +3.3 V	External TDM bus clock: provided by P4 to the MPC8260 and the QuadFALC.
8	CKIH	Tristate Output	Internal clock: generated by the QuadFALC and output on P4. This output is enabled only in "switched mode".
9			Not connected.
10			Not connected.

Table 5-6. PMC Connector P4 (cont)

No.	Pin Name	Pin Type	Description
11			Not connected.
12			Not connected.
13			Not connected.
14			Not connected.
15			Not connected.
16			Not connected.
17			Not connected.
18			Not connected.
19			Not connected.
20			Not connected.
21			Not connected.
22			Not connected.
23			Not connected.
24			Not connected.
25			Not connected.
26			Not connected.
27			Not connected.
28			Not connected.
29			Not connected.
30			Not connected.
31			Not connected.
32			Not connected.
33	RIN1_0	Analog Input	Line 0 Analog Input 1: The receive line presents a typical 100 Ω impedance.
34	ROUT1_0	Analog Output	Line 0 Analog Output 1.
35	RIN2_0	Analog Input	Line 0 Analog Input 2: The receive line presents a typical 100 Ω impedance.
36	ROUT2_0	Analog Output	Line 0 Analog Output 2.
37	RIN1_1	Analog Input	Line 1 Analog Input 1: The receive line presents a typical 100 Ω impedance.
38	ROUT1_1	Analog Output	Line 1 Analog Output 1.
39	RIN2_1	Analog Input	Line 1 Analog Input 2: The receive line presents a typical 100 Ω impedance.
40	ROUT2_1	Analog Output	Line 1 Analog Output 2.

Table 5-6. PMC Connector P4 (cont)

No.	Pin Name	Pin Type	Description
41	RIN1_2	Analog Input	Line 2 Analog Input 1: The receive line presents a typical 100 Ω impedance.
42	ROUT1_2	Analog Output	Line 2 Analog Output 1.
43	RIN2_2	Analog Input	Line 2 Analog Input 2: The receive line presents a typical 100 Ω impedance.
44	ROUT2_2	Analog Output	Line 2 Analog Output 2.
45	RIN1_3	Analog Input	Line 3 Analog Input 1: The receive line presents a typical 100 Ω impedance.
46	ROUT1_3	Analog Output	Line 3 Analog Output 1.
47	RIN2_3	Analog Input	Line 3 Analog Input 2: The receive line presents a typical 100 Ω impedance.
48	ROUT2_3	Analog Output	Line 3 Analog Output 2.
49			Not connected.
50			Not connected.
51			Not connected.
52			Not connected.
53			Not connected.
54			Not connected.
55			Not connected.
56			Not connected.
57			Not connected.
58			Not connected.
59			Not connected.
60			Not connected.
61			Not connected.
62			Not connected.
63			Not connected.
64			Not connected.

Debug Port J5

On the 4538, a 2x8-pin connector can be implemented to provide access to the BDM (Background Debug Mode) bus: the PowerQUICC II debug bus. Signals on this connector have 3.3V TTL electrical levels.

Table 5-7. J5 Debug Port

Pin	Signal	Signal	Pin
1	TDO	10 kΩ Pull-up to +3.3 V	2
3	TDI	TRST_N	4
5	QREQ_N	+3.3V through a 1 kΩ resistor	6
7	TCK		8
9	TMS		10
11	SRESET_N	GND	12
13	HRESET_N		14
15	10 kΩ Pull-up to +3.3 V	GND	16

WARNING

J5 Debug Connector is not compliant to PMC component height specification. It should be removed to insert the 4538 and its carrier into a CompactPCI chassis.

ISP Enable Jumper JP1

The 4538 includes a location for a jumper at JP1. This location is used during production to enable the programming of the card's EPLD programmable devices "in-situ". This location should never be used by the normal user.

Blank Card Jumper JP2

Location for a jumper at JP2 is needed for production when the PowerSPAN serial EEPROM is not yet programmed, in order to prevent the card from locking the system. This location should never be populated with a jumper by the normal user.

Connector Summary

Figure 5-5. 4538 Connectors

Connector	Usage
P1, P2	32-bit PCI bus
P4	Telecom Connector
J1	T1/E1/J1 line 0 (front panel)
J2	T1/E1/J1 line 1 (front panel)
J3	Ethernet 10/100 (front panel)
J4	RS232 TTY (front panel)
J5	JTAG/debug port
JP1	ISP programming
JP2	Blank card boot enable

Carrier Card Specification

CompactPCI Carrier Card

Interphase has defined a combination of cards to allow 4538 rear access configurations in CompactPCI chassis. The combination comprises a 4538 "rear access", a CPCI carrier card and an Interphase 6435 Rear Transition Module. The CPCI carrier card must be a Motorola CPV8540 or Motorola MCPN750 or a CPCI card conforming to the routing requirements described in Table 5-8 and Table 5-9. See also *PMC Connector P4* on page 103 for P4 connector compatibility.

NOTES

- **The Motorola cards will not allow the “Switched Mode” capability.**
- **Contact Interphase for additional information on Switched Mode compatibility**

Interconnections between CPCI J14 (PMC P4) and J3 are described in Table 5-8.

Interconnections between CPCI J24 (PMC P4) and J5 are described in Table 5-9.

Table 5-8. CompactPCI J3 Pin-Out

	ROW A	ROW B	ROW C	ROW D	ROW E	
14	+3.3V	+3.3V	+3.3V	+5V	+5V	14
13	PMC1IO5	PMC1IO4	PMC1IO3	PMC1IO2	PMC1IO1	13

Table 5-8. CompactPCI J3 Pin-Out (cont)

	ROW A	ROW B	ROW C	ROW D	ROW E	
12	PMC1IO10	PMC1IO9	*PMC1IO8*	PMC1IO7	*PMC1IO6*	12
11	*PMC1IO15*	*PMC1IO14*	PMC1IO13	*PMC1IO12*	PMC1IO11	11
10	PMC1IO20	PMC1IO19	*PMC1IO18*	PMC1IO17	*PMC1IO16*	10
9	PMC1IO25	PMC1IO24	PMC1IO23	PMC1IO22	PMC1IO21	9
8	PMC1IO30	PMC1IO29	PMC1IO28	PMC1IO27	PMC1IO26	8
7	**PMC1IO35**	**PMC1IO34**	**PMC1IO33**	PMC1IO32	PMC1IO31	7
6	**PMC1IO40**	**PMC1IO39**	**PMC1IO38**	**PMC1IO37**	**PMC1IO36**	6
5	**PMC1IO45**	**PMC1IO44**	**PMC1IO43**	**PMC1IO42**	**PMC1IO41**	5
4	*PMC1IO50*	*PMC1IO49*	**PMC1IO48**	**PMC1IO47**	**PMC1IO46**	4
3	*PMC1IO55*	*PMC1IO54*	*PMC1IO53*	*PMC1IO52*	*PMC1IO51*	3
2	PMC1IO60	PMC1IO59	PMC1IO58	PMC1IO57	PMC1IO56	2
1	V(I/O)	PMC1IO64	PMC1IO63	PMC1IO62	PMC1IO61	1

J3 Columns 15 to 19 are unused.

Table 5-9. CompactPCI J5 Pin-Out

	ROW A	ROW B	ROW C	ROW D	ROW E	
13	PMC2IO5 **(*)**	PMC2IO4 **(*)**	PMC2IO3 **(*)**	PMC2IO2 **(*)**	PMC2IO1 **(*)**	13
12	PMC2IO10	PMC2IO9	PMC2IO8 **(*)**	PMC2IO7 **(*)**	PMC2IO6 **(*)**	12
11	PMC2IO15	PMC2IO14	PMC2IO13	PMC2IO12	PMC2IO11	11
10	PMC2IO20	PMC2IO19	PMC2IO18	PMC2IO17	PMC2IO16	10
9	PMC2IO25	PMC2IO24	PMC2IO23	PMC2IO22	PMC2IO21	9
8	PMC2IO30	PMC2IO29	PMC2IO28	PMC2IO27	PMC2IO26	8
7	**PMC2IO35**	**PMC2IO34**	**PMC2IO33**	PMC2IO32	PMC2IO31	7
6	**PMC2IO40**	**PMC2IO39**	**PMC2IO38**	**PMC2IO37**	**PMC2IO36**	6
5	**PMC2IO45**	**PMC2IO44**	**PMC2IO43**	**PMC2IO42**	**PMC2IO41**	5
4	*PMC2IO50*	*PMC2IO49*	**PMC2IO48**	**PMC2IO47**	**PMC2IO46**	4
3	*PMC2IO55*	*PMC2IO54*	*PMC2IO53*	*PMC2IO52*	*PMC2IO51*	3
2	PMC2IO60	PMC2IO59	PMC2IO58	PMC2IO57	PMC2IO56	2
1	TMPRSNT	PMC2IO64	PMC2IO63	PMC2IO62	PMC2IO61	1

J5 Columns 14 to 19 are unused.

Signals printed in bold in Table 5-8 and Table 5-9 shall be routed to the corresponding PMC connector (used on 6435 RTM).

Signals printed in italics and underlined in Table 5-8 and Table 5-9 shall not be used on the carrier card. They should be routed to the corresponding PMC connector.

Signals followed by a **(*)** sign in Table 5-8 and Table 5-9 may be driven by the carrier card (No connection on the 6435 RTM). If switch mode is not supported by the carrier card, they should be routed to the corresponding PMC connector.

J5 A1 pin: TMRSNT: 6435 includes a 100 Ohm pull-up to 3.3 V (may be used by the carrier card to detect 6435 presence).

We recommend routing every other signal as indicated in Table 5-8 and Table 5-9 for future compatibility.

Custom Carrier Card

Customers who want to implement a 4538 "rear access" on configurations other than those described above will need to design their own line interfaces. For these customers, Interphase can provide additional information, such as schematics and a bill of material for these line interfaces.

6435 Rear Transition Module

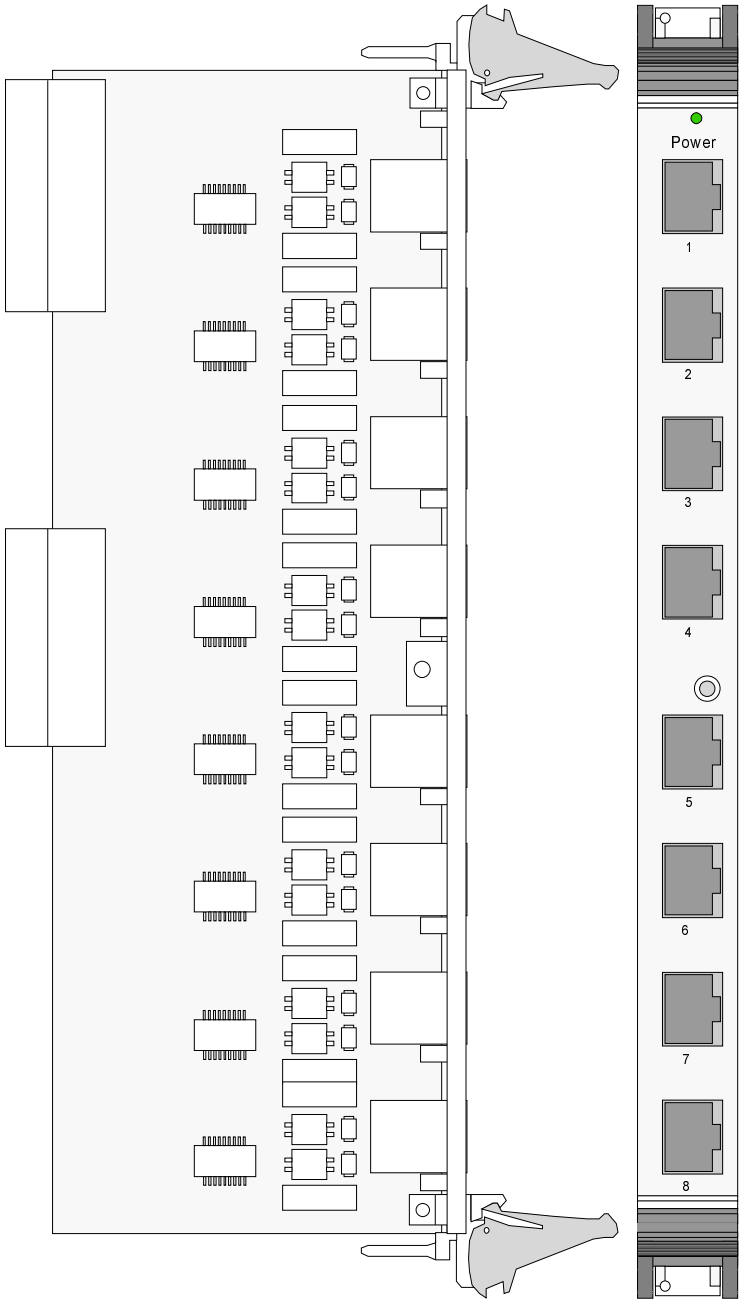


Figure 5-6. 8-Port 6435 Rear Transition Module Layout

NOTE

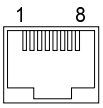
PMC site 1 (J14) corresponds to lines 5 to 8, PMC site 2 (J24) corresponds to line 1 to 4

The 6435 RTM is a fully passive module having the same features as the Front Panel T1/E1/J1 (described in *Front Panel* on page 96).

It includes eight T1/E1/J1 line interfaces with all the line safety protections. It does not include any loopback relays or line LEDs.

On the rear panel there is one unshielded 8-pin modular jack connector for each T1/E1/J1 line. The pin-out of the modular connectors follows the RJ48C definition as described in FCC part 68, and ISO /IEC 10173.

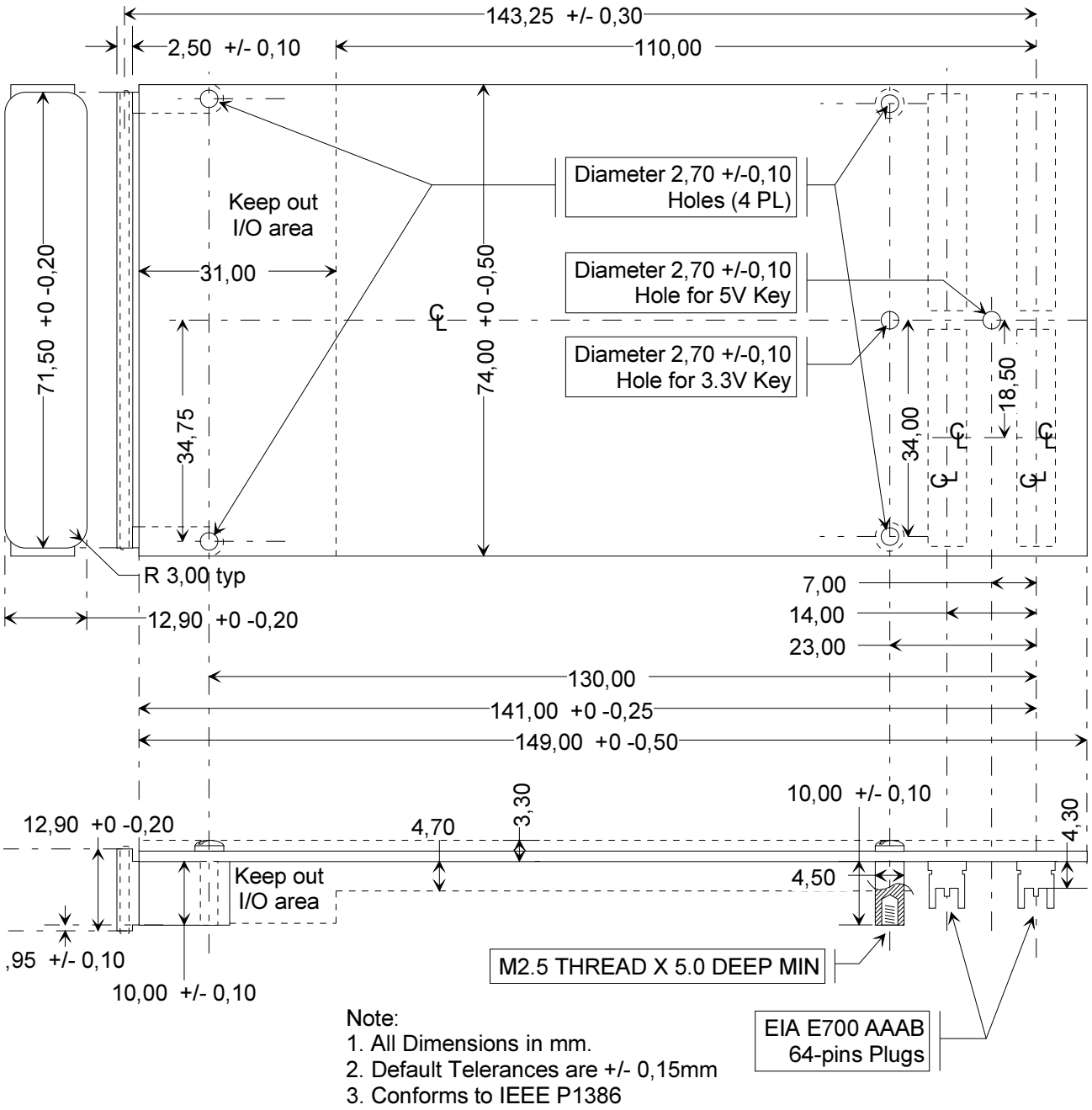
Table 5-10. T1/E1/J1 RJ48 Connector

	Signal
1	IN1
2	IN2
3	
4	OUT1
5	OUT2
6	
7	
8	

WARNING

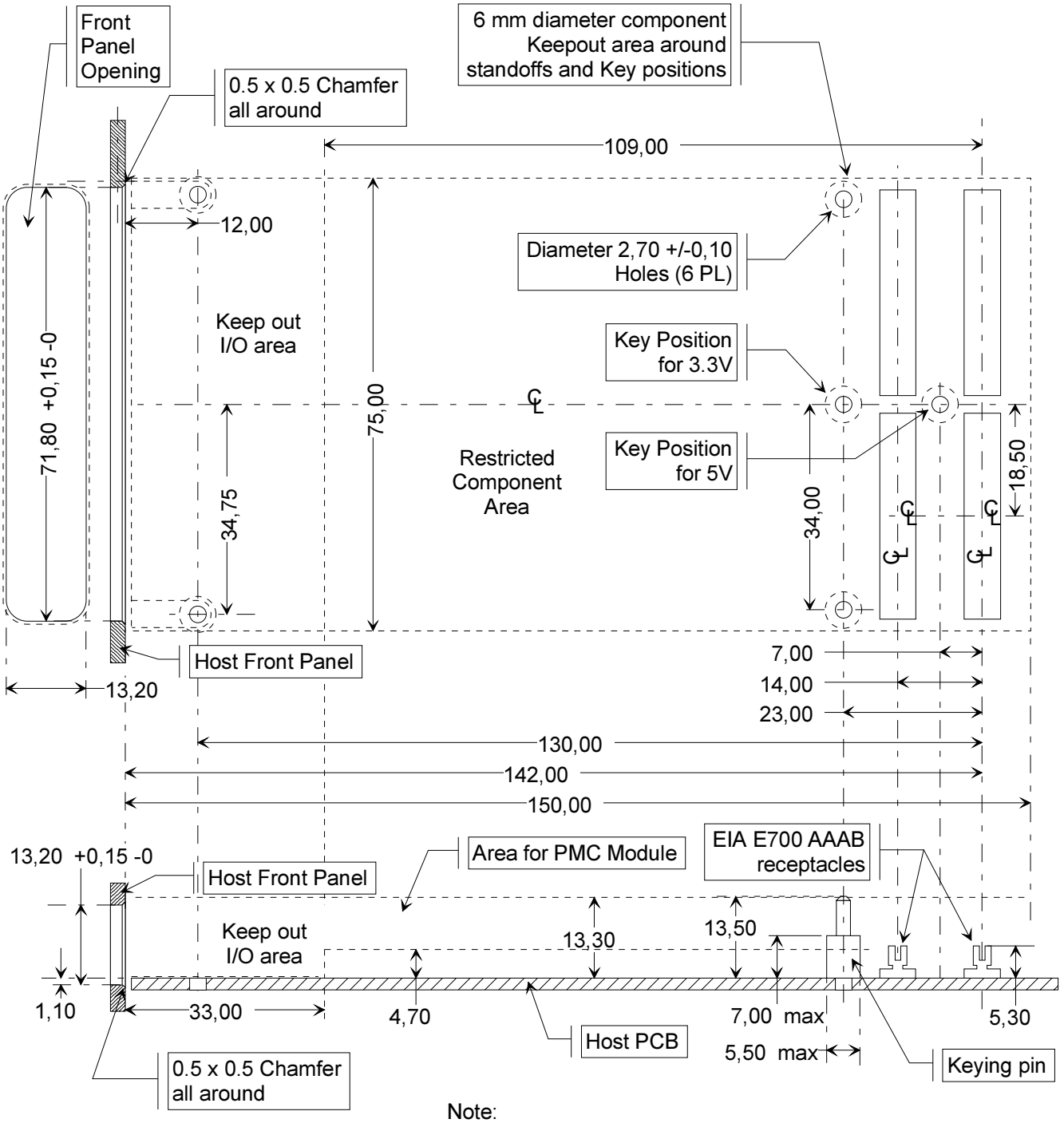
The 6435 RTM panel includes a green (power) LED, which indicates, when on, that removal of the board is NOT permitted.

PMC Card Dimensions



Note:
1. All Dimensions in mm.
2. Default Telerances are +/- 0,15mm
3. Conforms to IEEE P1386

Carrier Card Dimension Requirements



Note:
1. All Dimensions in mm.
2. Default Telerances are +/- 0,15mm
3. Conforms to IEEE P1386

Bibliography

Industry Standards

EIA-232-D: Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange

Electronic Industries Alliance
2500 Wilson Boulevard
Arlington, VA 22201-3834
Telephone: (703) 907-7500
Web: http://www.eia.org

ECTF H.110 Hardware Compatibility Specification: CT Bus, Revision 1.0

Enterprise Computer Telephony Forum
39355 California Street, Suite 307
Fremont CA 94538
Telephone: (510) 608-5915
Fax: (510) 608-5917
Web: http://www.ectf.org

IEEE Std 802.3, 2000, IEEE Standards for Local and Metropolitan Area Networks: Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T,

Institute of Electrical and Electronics Engineers, Inc.,
445 Hoes Lane, PO Box 1331
Piscataway, NJ 08855-1331
Telephone: (732) 981-0060
Web: http://www.ieee.com

PCI-SIG Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.2

PCI Special Interest Group
5200 N.E. Elam Young Parkway,
Hillsboro, OR 97124-6497
Telephone: 1-800-433-5177 or (503) 693-6232
FAX: (503) 693-8344
Web: http://www.pcisig.com

PICMG 2.0 CompactPCI Specification
PICMG 2.5 CompactPCI Computer Telephony Specification
PICMG 2.1 CompactPCI Hot Swap Specification

PCI Industrial Computer Manufacturers Group
401 Edgewater Place, Suite 500
Wakefield, MA 01880
Telephone: (781) 246-9318
Web: http:/www.picmg.org

PowerPC TM Microprocessor Common Hardware Reference Platform:

System Architecture (CHRP), Version 1.0
Literature Distribution Center for Motorola
Telephone: 1-800-441-2447
FAX: (602) 994-6430 or (303) 675-2150
E-mail: ldcformotorola@hibbertco.com
Publication reference: TB338/D

or
IBM
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6531
Telephone: 1-800-PowerPC
Publication reference: MPRP-CHRP-01

or
Morgan Kaufmann Publishers, Inc.
340 Pine Street, Sixth Floor
San Francisco, CA 94104-3205, USA
Telephone: (415) 392-2665 FAX: (415) 982-2665
Publication reference: ISBN 1-55860-394-8

PowerPC Reference Platform (PRP) Specification,
Third Edition, Version 1.0, Volumes I and II
International Business Machines Corporation
Power Personal Systems Architecture
11400 Burnet Rd.
Austin, TX 78758-3493
Document/Specification Ordering
Telephone: 1-800-PowerPC or 1-800-769-3772 or (708) 296-9332
Publication reference: MPR-PPC-RPU-02

Telecommunication Standards

ANSI T1.403-1995: Network-to-customer Installation - DS1 Metallic interface
ANSI T1.107-1995: Digital Hierarchy - Formats Specifications.
ANSI T1.102-1993: Digital Hierarchy - Electrical Interfaces.
ANSI T1.627-1993: Broadband ISDN - ATM Layer Functionality and Specification.

ANSI T1.107-1995: Digital Hierarchy - Formats Specifications.
ANSI T1.646-1995: Broadband ISDN - Physical Layer Specification for User-Network Interfaces Including DS1/ATM.

American National Standard for Telecommunications
ANSI American National Standard Institute
11 West 42nd Street
New York, New York 10036
T1 committee Web: http://www.t1.org

ATM Forum - ATM User-Network Interface Specification, V3.1, October, 1995.
ATM Forum - UTOPIA, An ATM PHY Interface Specification, Level 2, Version 1 June, 1995.

ATM forum
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Telephone (650) 949-6700
Fax: (650) 949-6705
http://www.atmforum.com

AT&T TR54016: Requirements for Interfacing Digital Terminal Equipment to Services Employing the Extended Superframe Format, Sept 89
AT&T TR62411: ACCUNET® T1.5 Service Description and Interface Specification, Dec. 90

AT&T Customer Information CenterHoward Press
1026 West Elizabeth Avenue
Linden, NJ 07036
Telephone: (888) 387-8852 (US order), (908) 523-2257 (international order)
Fax: (908)-862-6722
Web: http://www.att.com/cpetesting/trs.html

Bell Communications Research, TA-TSY-000773 - “Local Access System Generic Requirements, Objectives, and Interface in Support of Switched Multi-megabit Data Service” Issue 2, March 1990 and Supplement 1, December 1990.

Telcordia
Morris Corporate Center
tel : (973) 829-2000
Morristown, NJ
Web : http://www.telcordia.com

ETSI TBR 012 Ed1 (1993-12) Business Telecommunications (BT); Open Network Provision (ONP) technical requirements; 2 048 kbit/s digital unstructured leased line (D2048U) Attachment requirements for terminal equipment interface.
ETSI TBR 013 (1996-01) Business TeleCommunications (BTC); 2 048 kbit/s digital structured leased lines (D2048S); Attachment requirements for terminal equipment interface.
ETSI ETS 300 269 Draft Standard T/NA(91)17 - Metropolitan Area Network Physical Layer Convergence Procedure for 2.048 Mbit/s”, April 1994.
ETSI ETS 300 011 - Integrated Services Digital Network (ISDN); Primary rate user-network interface Layer

1 specification and test principles.
ETSI ETS 300 166 - Transmission and Multiplexing (TM); Physical and electrical characteristics of hierarchical digital interfaces for equipment using the 2 048 kbit/s - based plesiochronous or synchronous digital hierarchies.
ETSI ETS 300 233 - Integrated Services Digital Network (ISDN); Access digital section for ISDN primary rate.

European Telecommunications Standards Institute (ETSI)
Sophia Antipolis
http://www.etsi.org/

ITU-T G.703 - Physical/Electrical Characteristics of Hierarchical Digital Interfaces, 1991 (includes 2.048Mbps E1 and 1.56Mbps T1 definitions)
ITU-T G.704 - Terminal Equipment Synchronous Frame Structures Used At 1544, 6312, 2048, 8488 and 44 736 kbit/s Hierarchical Levels", July, 1995.
ITU-T I.431 - Primary rate user-network interface - Layer 1 specification, 1993
ITU-T I.432 - B-ISDN User-Network Interface - Physical Layer Specification, 1993
ITU-T G.804 - ATM Cell Mapping into Plesiochronous Digital Hierarchy (PDH), 1993.
ITU-T G.832 - Transport of SDH Elements on PDH Networks: Frame and Multiplexing Structures", 1993.
ITU-T Q.921 - ISDN User-Network Interface - Data Link Layer Specification, March, 1993.
Other ITU-T references: G.705, G.706, G.732, G.735, G.736, G.737, G.738, G.739, G.751, G775, G.823, G.824, O.151

International Telecommunications Union (ITU)
Place des Nations
CH-1211 Geneva 20
Switzerland
Telephone: +41-22-730-5111
Web: http://www.itu.int/ITU-T

Japanese references: JT- G.703, JT G.704, JT G.706

Manufacturers' Documents

PowerQUICC II Literature

MPC8260 PowerQUICC II User's Manual
Literature Distribution Center for Motorola
Telephone: 1-800-441-2447
FAX: (602) 994-6430 or (303) 675-2150
E-mail: ldcformotorola@hibbertco.com
http://www.mot-sps.com/

Tundra PowerSpan PCI Bridge

PowerSpan™
(CA91L8200/CA91L8260)
PowerPC/PCI Bridge
User Manual
80A1000_MA001_08
http://www.tundra.com

Infineon PEB22554 / QuadFALC T1/E1/J1 framer

Quad Framing and Line Interface Component for E1 / T1 / J1
QuadFALC TM
PEB 22554 Version 1.3
Data Sheet 07.2000
Infineon Technologies AG,
P.O. Box 800949,
81609 Munich, Germany
http://www.infineon.com/products/commics/QuadFALC.htm

Intel LXT971A - 3.3V Dual-Speed Fast Ethernet Transceiver

Intel Corporation (Santa Clara)
2200 Mission College Blvd.
Santa Clara, California 95052-8119
USA Tel: 408-765-8080 Fax: 408-765-9904
http://www.intel.com

Motorola CompactPCI CPV8540B and MCPN750 carrier boards

Corporate Headquarters Motorola Computer Group
2900 South Diablo Way
Tempe, AZ 85282
Tel: (800) 759-1107
http://www.mot.com/computer

Glossary

AAL ◆ **ATM Adaptation Layer** Service-dependent sublayer of the data link layer. The AAL accepts data from different applications and presents it to the *ATM* layer in the form of 48-byte ATM payload segments. AALs consist of two sublayers: *CS* and *SAR*. AALs differ on the basis of the source-destination timing used, whether they use *CBR* or *VBR*, and whether they are used for connection-oriented or connectionless mode data transfer. At present, the four types of AAL recommended by the ITU-T are *AAL1*, *AAL2*, *AAL3/4*, and *AAL5*.

AAL1 ◆ **ATM Adaptation Layer 1**One of four AALs recommended by the ITU-T. AAL1 is used for connection-oriented, delay-sensitive services requiring constant bit rates, such as uncompressed video and other isochronous traffic.

AAL2 ◆ **ATM Adaptation Layer 2** One of four AALs recommended by the ITU-T. AAL2 is used for connection-oriented services that support a variable bit rate, such as some isochronous video and voice traffic.

AAL3/4 ◆ **ATM Adaptation Layer 3/4** One of four AALs (merged from two initially distinct adaptation layers) recommended by the ITU-T. AAL3/4 supports both connectionless and connection-oriented links, but is primarily used for the transmission of *SMDS* packets over *ATM* networks.

AAL5 ◆ **ATM Adaptation Layer 5** One of four AALs recommended by the ITU-T. AAL5 supports connection-oriented *VBR* services and is used predominantly for the transfer of classical *IP* over *ATM* and *LANE* traffic. AAL5 uses *SEAL* and is the least complex of the current AAL recommendations. It offers low bandwidth overhead and simpler processing requirements in exchange for reduced bandwidth capacity and error-recovery capability.

AIN ◆ **Advanced Intelligent Network** In *SS7*, an expanded set of network services made available to the user, and under user control, that requires improvement in network switch architecture, signaling capabilities, and peripherals.

AMI ◆ **Alternate Mark Inversion** Line-code type used on *T1* and *E1* circuits. In AMI, zeros are represented by 01 during each bit cell, and ones are represented by 11 or 00, alternately, during each bit cell. AMI requires that the sending device maintain ones density. Ones density is not maintained independently of the data stream. Sometimes called binary coded alternate mark inversion.

API ◆ **Application Programming Interface** (1) The interface to a library of language-specific subroutines (such as a graphics library) that implement higher-level functions. (2) A set of calling conventions defining how a service is invoked through a software package.

ASCII ◆ **American Standard Code for Information Interchange** The standard binary encoding of alphabetical characters, numbers, and other keyboard symbols.

ATM ◆ **Asynchronous Transfer Mode** International standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as *E3*, *SONET*, and *T3*.

B8ZS ◆ **Binary 8-Zero Substitution** Line-code type, used on *T1* and *E1* circuits, in which a special code is substituted whenever 8 consecutive zeros are sent over the link. This code is then interpreted at the remote end of the connection. This technique guarantees ones density independent of the data stream. Sometimes called bipolar 8-zero substitution.

B Channel ◆ **Bearer Channel** In ISDN, a full-duplex, 64-kbps channel used to send user data.

BIOS ◆ **Basic Input/Output System** The built-in program that controls the basic functions of communications between the processor and the Input/Output (I/O) devices of a computer.

BISDN ◆ **Broadband ISDN** *ITU-T* communication standards designed to handle high-bandwidth applications such as video. BISDN currently uses *ATM* technology over *SONET*-based transmission circuits to provide data rates from 155 to 622 Mbps and beyond.

bootROM ◆ **boot Read-Only Memory** Chip mounted on the printed circuit board used to provide executable boot instructions to a computer device.

BRI ◆ **Basic Rate Interface** *ISDN* interface composed of two *B Channel*s and one *D Channel* for circuit-switched communication of voice, video, and data.

BSP ◆ **Board Support Package** A board support package consists of documentation and software used to configure and install a specific operating system on a specific product.

BUS ◆ **Broadcast and Unknown Server** Multicast server used in *ELAN*s that is used to flood traffic addressed to an unknown destination and to forward multicast and broadcast traffic to the appropriate clients.

CAM ◆ **Content Addressable Memory** Memory that is accessed based on its contents, not on its memory address.

CBR ◆ **Constant Bit Rate** *QoS* class defined by the *ATM* Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery.

CCS ◆ **Common Channel Signaling** Signaling system used in telephone networks that separates signaling information from user data. A specified channel is exclusively designated to carry signaling information for all other channels in the system.

COM ◆ **Communication or Communications**

CompactPCI ◆ CompactPCI is an adaptation of the Peripheral Component Interconnect (*PCI*) Specification for industrial and/or embedded applications requiring a more robust mechanical form factor than desktop PCI. CompactPCI uses industry standard mechanical components and high performance connector technologies to provide an optimized system intended for rugged applications. CompactPCI provides a system that is electrically compatible with the PCI Specification, allowing low cost PCI components to be utilized in a mechanical form factor suited for rugged environments. CompactPCI is an open specification supported by the PICMG (PCI Industrial Computer Manufacturers Group), which is a consortium of companies involved in utilizing PCI for embedded applications.

CPCS ◆ **Common Part Convergence Sublayer** An abstract *ATM* protocol *API* defined by the *ATM* Forum. It forms the boundary interface between the purely software implemented higher layer ATM protocols and the segmentation and reassembly process controlled by hardware.

CPM ◆ **Communication Processing Module**

CRC4 ◆ **Cyclic Redundancy Check**. Error-checking technique in which the frame recipient calculates a remainder by dividing frame contents by a prime binary divisor and compares the calculated remainder to a value stored in the frame by the sending node.

CS ◆ **Convergence Sublayer** One of the two sublayers of the *AAL CPCS*, which is responsible for padding and error checking. *PDU*s passed from the *SSCS* are appended with an 8-byte trailer (for error checking and other control information) and padded, if necessary, so that the length of the resulting PDU is divisible by 48. These PDUs are then passed to the *SAR* sublayer of the CPCS for further processing.

CSU ◆ **Channel Service Unit** A component that terminates a digital circuit, such as *T1*. A CSU assures compliance to FCC regulations and preforms some line-conditioning functions.

D Channel ◆ **Data Channel** Full-duplex, 16-kbps (*BRI*) or 64-kbps (*PRI*) *ISDN* channel.

DCE ◆ 1. **Data Communications Equipment** (EIA expansion). 2. **Data Circuit-terminating Equipment** (ITU-T expansion). Devices and connections of a communications network that comprise the network end of the user-to-network interface. The DCE provides a physical connection to the network, forwards traffic, and provides a clocking signal used to synchronize data transmission between DCE and DTE devices. Modems and interface cards are examples of DCE.

DLCI ◆ **Data-Link Connection Identifier** Value that specifies a *PVC* or *SVC* in a *Frame Relay* network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the *LMI* extended specification, DLCIs are globally significant (DLCIs specify individual end devices).

DMA ◆ **Direct Memory Access** The transfer of data directly into memory without supervision of the processor. The data is passed on the bus directly between the memory and another device.

DPRAM ◆ **Dual Port Random Access Memory**

DS1 ◆ **Digital Signal level 1** Framing specification used in transmitting digital signals at 1.544-Mbps on a *T1* facility (in the United States) or at 2.108-Mbps on an *E1* facility (in Europe).

DS3 ◆ **Digital Signal level 3** Framing specification used for transmitting digital signals at 44.736 Mbps on a *T3* facility.

DSX1 ◆ Cross-connection point for *DS1* signals.

DTE ◆ **Data Terminal Equipment** Device at the user end of a user-network interface that serves as a data source, destination, or both. DTE connects to a data network through a *DCE* device (for example, a modem) and typically uses clocking signals generated by the DCE. DTE includes such devices as computers, protocol translators, and multiplexers.

E1 ◆ Wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 2.048 Mbps. E1 lines can be leased for private use from common carriers.

E3 ◆ Wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 34.368 Mbps. E3 lines can be leased for private use from common carriers.

EEPROM ◆ **Electrically Erasable Programmable Read-Only Memory** A nonvolatile *PROM* that can be written as well as read form. Usually used to hold information about the current system configuration, alternate boot paths, etc.

ELAN ◆ **Emulated LAN** *ATM* network in which an Ethernet or Token Ring *LAN* is emulated using a client-server model. ELANs are composed of an *LEC*, an *LES*, a *BUS*, and an *LECS*. Multiple ELANs can exist simultaneously on a single ATM network. ELANs are defined by the *LANE* specification.

END ◆ **Enhanced Network Driver**

EPLD ◆ **Electrically Programmable Logic Device**

ES ◆ **End System** Generally, an end-user device on a network.

ESF ◆ **Extended Superframe Format** Framing type used on *T1* circuits that consists of 24 frames of 192 bits each, with the 193rd bit providing timing and other functions.

Ethernet ◆ Baseband *LAN* specification invented by Xerox Corporation and developed jointly by Xerox, Intel, and Digital Equipment Corporation.

FCC ◆ **Federal Communications Commission** The Government agency responsible for regulating telecommunications in the United States.

FCC ◆ **Fast serial Communication Controllers** Used to control the fast Ethernet port.

FDDI ◆ **Fiber Distributed Data Interface** *LAN* standard, defined by ANSI X3T9.5, specifying a 100-Mbps token-passing network using fiber-optic cable, with transmission distances of up to 2 km. FDDI uses a dual-ring architecture to provide redundancy.

Flash ◆ Nonvolatile storage that can be electrically erased and reprogrammed so that software images can be stored, booted, and rewritten as necessary.

Frame Relay ◆ Industry-standard, switched data link layer protocol that handles multiple virtual circuits using *HDLC* encapsulation between connected devices. Frame Relay is more efficient than *X.25*, the protocol for which it is generally considered a replacement.

FTP ◆ **File Transfer Protocol** Application protocol, part of the *TCP/IP* protocol stack, used for transferring files between network nodes.

GB ◆ **GigaBytes** 10^9 bytes per second.

Gbps ◆ **Gigabits per second** 10^9 bits per second.

HDLC ◆ **High-Level Data Link Control** Bit-oriented synchronous data link layer protocol developed by *ISO*. Derived from *SDLC*, *HDLC* specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

IMA ◆ **Inverse Multiplexing over ATM** Standard protocol defined by the ATM Forum in 1997.

IMMR ◆ **Internal Memory Map Register**

IP ◆ **Internet Protocol** Network layer protocol in the *TCP/IP* stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security.

ISDN ◆ **Integrated Services Digital Network** Communication protocol, offered by telephone companies, that permits telephone networks to carry data, voice, and other source traffic.

ISO ◆ **International Organization for Standardization** International organization that is responsible for a wide range of standards, including those relevant to networking. ISO developed the *OSI* reference model, a popular networking reference model.

ITU-T ◆ **International Telecommunication Union Telecommunication Standardization Sector** International body that develops worldwide standards for telecommunications technologies. The ITU-T carries out the functions of the former CCITT.

J1 ◆ Japanese transmission standard

LAN ◆ **Local-Area Network** High-speed, low-error data network covering a relatively small geographic area (up to a few thousand meters). LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited area. LAN standards specify cabling and signaling at the physical and data link layers of the *OSI* model. *Ethernet*, *FDDI*, and *Token Ring* are widely used LAN technologies.

LANE ◆ **LAN Emulation** Technology that allows an *ATM* network to function as a *LAN* backbone. The ATM network must provide multicast and broadcast support, address mapping (*MAC Address*-to-ATM), *SVC* management, and a usable packet format. LANE also defines *Ethernet* and *Token Ring ELAN*s.

LAPB ◆ **Link Access Procedure, Balanced**. Data link layer protocol in the *X.25* protocol stack. LAPB is a bit-oriented protocol derived from *HDLC*.

LEC ◆ **LAN Emulation Client** Entity in an end system that performs data forwarding, address resolution, and other control functions for a single *ES* within a single *ELAN*. An LEC also provides a standard *LAN* service interface to any higher-layer entity that interfaces to the LEC. Each LEC is identified by a unique *ATM* address, and is associated with one or more *MAC Address*es reachable through that ATM address.

LECS ◆ **LAN Emulation Configuration Server** Entity that assigns individual *LANE* clients to particular *ELAN*s by directing them to the *LES* that corresponds to the ELAN. There is logically one LECS per administrative domain, and this serves all ELANs within that domain.

LED ◆ **Light Emitting Diode** A semiconductor device used to provide visual indications, used in place of an incandescent light. Also a semiconductor device used to transmit light into a fiber.

LES ◆ **LAN Emulation Server** Entity that implements the control function for a particular *ELAN*. There is only one logical LES per ELAN, and it is identified by a unique *ATM* address.

LMI ◆ **Local Management Interface** Set of enhancements to the basic *Frame Relay* specification. LMI includes support for a keepalive mechanism, which verifies that data is flowing; a multicast mechanism, which provides the network server with its local *DLCI* and the multicast DLCI; global addressing, which gives DLCIs global rather than local significance in Frame Relay networks; and a status mechanism, which provides an on-going status report on the DLCIs known to the switch. Known as LMT in ANSI terminology.

MAC Address ◆ Standardized data link layer address that is required for every port or device that connects to a *LAN*. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, MAC-layer address, and physical address.

MCC ◆ **Multichannel Communication Controller**

MiniDIN ◆ Miniature multi-pin connector.

MUX ◆ **Multiplexer** Combines multiple signals for transmission over a single line. The signals are demultiplexed, or separated, at the receiving end

NT1 ◆ **Network Termination 1** A device that provides the interface between customer premises equipment and central office switching equipment.

NVRAM ◆ **Nonvolatile RAM** *RAM* that retains its contents when a unit is powered off.

OC3 ◆ **Optical Carrier 3** Physical protocol defined for *SONET* optical signal transmissions. OC3 signal levels put *STS* frames onto multimode fiber-optic line at at 155.52 Mbps.

OSI ◆ **Open System Interconnection** International standardization program created by *ISO* and *ITU-T* to develop standards for data networking that facilitate multivendor equipment interoperability.

PCI ◆ **Peripheral Component Interconnect** A high-performance multiplexed address and data bus. Supporting 32-bit with optional 64-bit data transfers, the PCI bus is intended to be an interconnect between peripheral controllers, peripheral add-in boards, and processor/memory systems. The PCI bus operates at up to 33 MHz, providing burst transfer rates up to 132 MBps 32 bits wide, or up to 264 MBps 64 bits wide.

PDN ◆ **Public Data Network** Network operated either by a government (as in Europe) or by a private concern to provide computer communications to the public, usually for a fee. PDNs enable small organizations to create a *WAN* without all the equipment costs of long-distance circuits.

PDU ◆ Protocol Data Unit A message of a given protocol comprising payload and protocol-specific control information, typically contained in a header.

PLP ◆ Packet Level Protocol Network layer protocol in the *X.25* protocol stack. Sometimes called X.25 Level 3 and X.25 Protocol.

PMC ◆ PCI Mezzanine Card *PCI* "daughter" card designed to mount on a "mother card".

POST ◆ Power-On-Self-Test Test that automatically runs whenever the power is applied to the card.

PRI ◆ Primary Rate Interface *ISDN* interface to primary rate access. Primary rate access consists of a single 64-Kbps *D Channel* plus 23 (*T1*) or 30 (*E1*) *B Channel*s for voice or data.

PROM ◆ Programmable Read-Only Memory *ROM* that can be programmed using special equipment. PROMs can be programmed only once.

PVC ◆ Permanent Virtual Circuit or Connection Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In *ATM* terminology, called a permanent virtual connection.

QoS ◆ Quality of Service Measure of performance for a transmission system that reflects its transmission quality and service availability.

RAM ◆ Random-Access Memory Volatile memory that can be read and written by a microprocessor.

RISC ◆ Reduced Instruction Set Computing

ROM ◆ Read-Only Memory Nonvolatile memory that can be read, but not written, by the microprocessor.

RTM ◆ Rear Transition Module A module that provides network connections from the rear of a system.

Rx ◆ Receive or Receiver

SAR ◆ Segmentation And Reassembly One of the two sublayers of the *AAL CPCS*, responsible for dividing (at the source) and reassembling (at the destination) the *PDU*s passed from the *CS*. The SAR sublayer takes the PDUs processed by the CS and, after dividing them into 48-byte pieces of payload data, passes them to the *ATM* layer for further processing.

SCC ◆ Serial Communication Controller

SDH ◆ Synchronous Digital Hierarchy European standard that defines a set of rate and format standards that are transmitted using optical signals over fiber. SDH is similar to *SONET*, with a basic SDH rate of 155.52 Mbps, designated at *STM-1*.

SDLC ◆ Synchronous Data Link Control *SNA* data link layer communications protocol. SDLC is a bit-oriented, full-duplex serial protocol that has spawned numerous similar protocols, including *HDLC* and *LAPB*.

SDU ◆ Service Data Unit A unit of interface information whose identity is preserved from one end of a layer connection to the other.

SDRAM ◆ Synchronous Digital Random Access Memory

SEAL ◆ Simple And Efficient AAL Scheme used by *AAL5* in which the *SAR* sublayer segments *CS PDU*s without adding additional fields.

SIU ◆ Serial Interface Unit

SMC ◆ Serial Management Controller

SMDS ◆ Switched Multimegabit Data Service High-speed, packet-switched, datagram-based *WAN* networking technology offered by the telephone companies.

SNA ◆ Systems Network Architecture Large, complex, feature-rich network architecture developed in the 1970s by IBM.

SONET ◆ Synchronous Optical Network High-speed (up to 2.5 Gbps) synchronous network specification developed by Bellcore and designed to run on optical fiber. *STS1* is the basic building block of SONET. Approved as an international standard in 1988.

SS7 ◆ Signaling System 7 Standard *CCS* system used with *BISDN* and *ISDN*.

SSCS ◆ Service Specific Convergence Sublayer One of the two sublayers of any *AAL*. SSCS, which is service dependent, offers assured data transmission. The SSCS can be null as well, in classical *IP* over *ATM* or *LAN* emulation implementations.

STM-1 ◆ Synchronous Transport Module level 1 One of a number of *SDH* formats that specifies the frame structure for the 155.52-Mbps lines used to carry *ATM* cells.

STS ◆ **Synchronous Transport Signal**

STS1 ◆ **Synchronous Transport Signal level 1** Basic building block signal of *SONET*, operating at 51.84 Mbps. Faster SONET rates are defined as STS-n, where n is a multiple of 51.84 Mbps.

SVC ◆ **Switched Virtual Circuit** Virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic. Called a switched virtual connection in *ATM* terminology.

T1 ◆ T1 transmits *DS1*-formatted data at 1.544 Mbps through the telephone-switching network, using *AMI* or *B8ZS* coding.

T3 ◆ Digital *WAN* carrier facility. T3 transmits *DS3*-formatted data at 44.736 Mbps through the telephone switching network.

TCP ◆ **Transmission Control Protocol** Connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the *TCP/IP* protocol stack.

TCP/IP ◆ **Transmission Control Protocol/Internet Protocol** Common name for the suite of protocols developed by the U.S. DoD in the 1970s to support the construction of worldwide internetworks. *TCP* and *IP* are the two best-known protocols in the suite.

TFTP ◆ **Trivial File Transfer Protocol** Simplified version of *FTP* that allows files to be transferred from one computer to another over a network.

Token Ring ◆ Token-passing *LAN* developed and supported by IBM. Token Ring runs at 4 or 16 Mbps over a ring topology. Similar to IEEE 802.5.

TTY ◆ **Teletypewriter** General term for an input device.

Tx ◆ **Transmit or Transmitter**

USRBUF ◆ A driver structure describing the use of a specific buffer containing payload data to be transferred using *ATM*. They can be linked together to allow non-contiguous areas of memory to be sent as one unit.

X.25 ◆ *ITU-T* standard that defines how connections between *DTE* and *DCE* are maintained for remote terminal access and computer communications in *PDN*s. X.25 specifies *LAPB*, a data link layer protocol, and *PLP*, a network layer protocol. *Frame Relay* has to some degree superseded X.25.

VBR ◆ **Variable Bit Rate** *QoS* class defined by the *ATM* Forum for ATM networks. VBR is subdivided into a Real Time (RT) class and Non-Real Time (NRT) class. VBR (RT) is used for connections in which there is a fixed timing relationship between samples. VBR (NRT) is used for connections in which there is no fixed timing relationship between samples, but that still need a guaranteed QoS.

VCC ◆ **Virtual Channel Connection** Can be a Permanent Virtual Connection (*PVC*) or a Switched Virtual Connection (*SVC*). Any *ATM* connection between two nodes.

VCI ◆ **Virtual Channel Identifier**16-bit field in the header of an *ATM* cell. The VCI, together with the *VPI*, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next network VCL that a cell needs to transit on its way to its final destination. The function of the VCI is similar to that of the *DLCI* in *Frame Relay*.

VCL ◆ **Virtual Channel Link** Connection between two *ATM* devices. A *VCC* is made up of one or more VCLs.

VPI ◆ **Virtual Path Identifier** 8-bit field in the header of an *ATM* cell. The VPI, together with the *VCI*, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next VCL that a cell needs to transit on its way to its final destination. The function of the VPI is similar to that of the *DLCI* in *Frame Relay*.

WAN ◆ **Wide-Area Network** Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers. *Frame Relay*, *SMDS*, and *X.25* are examples of WANs.

Index

When using this index, keep in mind that a page number indicates only where referenced material begins. It may extend to the page or pages following the page referenced.