



PLXMon 99
Software Development Utility
User's Manual

Version 3.0

© 1999 PLX Technology, Inc. All rights reserved.

Copyright © 1999, PLX Technology, Inc.. All rights reserved.

This document contains proprietary and confidential information of PLX Technology Inc. (PLX). The contents of this document may not be copied nor duplicated in any form, in whole or in part, without prior written consent from PLX Technology, INC..

PLX provides the information and data included in this document for your benefit, but it is not possible for us to entirely verify and test all of this information in all circumstances, particularly information relating to non-PLX manufactured products. PLX makes no warranties or representations relating to the quality, content or adequacy of this information. Every effort has been made to ensure the accuracy of this manual, however, PLX assumes no responsibility for any errors or omissions in this document. PLX shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. PLX assumes no responsibility for any damage or loss resulting from the use of this manual; for any loss or claims by third parties which may arise through the use of this SDK; and for any damage or loss caused by deletion of data as a result of malfunction or repair. The information in this document is subject to change without notice.

Product and Company names are trademarks or registered trademarks of their respective owners.

Document number: PLXMON99-SDK-MAN-P1-3.0



PLX Software License Agreement

PLX SOFTWARE LICENSE AGREEMENT

THIS PLX SOFTWARE IS LICENSED TO YOU UNDER SPECIFIC TERMS AND CONDITIONS. CAREFULLY READ THE TERMS AND CONDITIONS PRIOR TO USING THIS SOFTWARE. OPENING THIS SOFTWARE PACKAGE OR INITIAL USE OF THIS SOFTWARE INDICATES YOUR ACCEPTANCE OF THE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD RETURN THE ENTIRE SOFTWARE PACKAGE TO PLX.

LICENSE Copyright (c) 1999 PLX Technology, Inc.

This PLX Software License agreement is a legal agreement between you and PLX Technology, Inc. for the PLX Software, which is provided on the enclosed PLX CD-ROM. PLX Technology owns this PLX Software. The PLX Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties, and is licensed, not sold. If you are a rightful possessor of the PLX Software, PLX grants you a license to use the PLX Software as part of or in conjunction with a PLX chip on a **per project basis**. PLX grants this permission provided that the above copyright notice appears in all copies and derivatives of the PLX Software. Use of any supplied runtime object modules or derivatives from the included source code in any product without a PLX Technology, Inc. chip is strictly prohibited. You obtain no rights other than those granted to you under this license. You may copy the PLX Software for backup or archival purposes. You are not authorized to use, merge, copy, display, adapt, modify, execute, distribute or transfer, reverse assemble, reverse compile, decode, or translate the PLX Software except to the extent permitted by law.

GENERAL

If you do not agree to the terms and conditions of this PLX Software License Agreement, do not install or

use the PLX Software and promptly return the entire unused PLX Software to PLX Technology, Inc. You may terminate your PLX Software license at any time. PLX Technology may terminate your PLX Software license if you fail to comply with the terms and conditions of this License Agreement. In either event, you must destroy all your copies of this PLX Software. Any attempt to sub-license, rent, lease, assign or to transfer the PLX Software except as expressly provided by this license, is hereby rendered null and void.

WARRANTY

PLX Technology, Inc. provides this PLX Software AS IS, WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. PLX makes no guarantee or representations regarding the use of, or the results based on the use of the software and documentation in terms of correctness, or otherwise; and that you rely on the software, documentation, and results solely at your own risk. In no event shall PLX be liable for any loss of use, loss of business, loss of profits, incidental, special or, consequential damages of any kind. In no event shall PLX's total liability exceed the sum paid to PLX for the product licensed here under.

PLX Copyright Message Guidelines

The following copyright message along with the following text must appear in all software products generated and distributed which use the PLX API libraries:

“Copyright (c) 1999 PLX Technology, Inc.”

Requirements:

- Arial font
- Font size 12 (minimum)
- Bold type
- Must appear as shown above in the first section or the so called “Introduction Section” of all manuals
- Must also appear as shown above in the beginning of source code as a comment

This page intentionally left blank.

TABLE OF CONTENTS

1. Introduction.....	1-1
1.1 About This Manual	1-1
1.2 Conventions And Support	1-1
1.3 Installation	1-2
1.4 PLXMon 99 Feature List	1-2
1.5 Customer Support Information	1-3
2. A Brief Tour of PLXMon 99	2-1
2.1 Starting PLXMon 99 And Configuration	2-1
2.1.1 <i>Default Vendor and Device IDs for PLX Reference Design Boards</i>	2-3
2.2 Displaying Registers	2-3
2.3 Popular Features	2-3
2.3.1 <i>EEPROM Edit Utility</i>	2-4
2.3.2 <i>Downloading To The IOP</i>	2-4
2.3.3 <i>DMA Transfers</i>	2-5
2.4 Board with No Local CPU	2-7
2.4.1 <i>Using PCI 9054RDK-Lite Board or Your Custom Board with No CPU</i>	2-7
3. PLXMon 99 - Reference	3-1
3.1 Access Mode	3-1
3.1.1 <i>PCI Mode</i>	3-1
3.1.2 <i>IOP (Serial) Mode</i>	3-1
3.2 Application Hot Links	3-2
3.3 Device Configuration	3-2
3.4 Downloading IOP Applications	3-3
3.5 Font Configuration	3-4
3.6 The Interface	3-4
3.6.1 <i>The PLXMon 99 Toolbar</i>	3-5
3.6.2 <i>Status Bar</i>	3-5
3.6.3 <i>Command Line Interface (CLI)</i>	3-5
3.6.3.1 <i>Displaying Memory Via Memory Cycles, (dl, dw, db)</i>	3-5
3.6.3.2 <i>Displaying Memory Via I/O Cycles, (il, iw, ib)</i>	3-5
3.6.3.3 <i>Writing Memory Via Memory Cycles, (el, ew, eb)</i>	3-6
3.6.3.4 <i>Writing Memory Via I/O Cycles, (ol, ow, ob)</i>	3-6
3.6.3.5 <i>The Pci Command</i>	3-6
3.6.3.6 <i>The Quit Command</i>	3-6
3.6.3.7 <i>The Reg Command</i>	3-7
3.6.3.8 <i>The Repeat Command (r)</i>	3-7
3.6.3.9 <i>User Variables (vars)</i>	3-7
3.6.3.10 <i>The Ver Command</i>	3-7
3.7 Print, Print Preview, and Print Setup	3-7
3.8 Register Access/Register Sets	3-8
3.9 The Reset Button	3-8
3.10 Selecting Devices	3-8
3.11 Serial Configuration	3-9
3.12 Serial EEPROM Access	3-9
4. The IOP 480 Register Set	4-1
4.1 The Register Group Dialog Boxes	4-1
4.1.1 <i>PCI Configuration Register Group Dialog Box</i>	4-1
4.1.2 <i>Local Configuration Register Group Dialog Box</i>	4-2
4.1.3 <i>The Runtime Register Group Dialog Box</i>	4-2
4.1.4 <i>The DMA Register Group Dialog Box</i>	4-3



4.1.5	The Messaging Queue Register Group Dialog Box	4-4
4.1.6	IOP 480 Memory Controller register Group Dialog Box	4-5
4.1.7	IOP 480 CPU Registers Group Dialog Box.....	4-6
4.1.8	IOP 480 EEPROM Values	4-7
5.	The PCI 9054 Register Set.....	5-1
5.1	The Register Group Dialog Boxes	5-1
5.1.1	PCI Configuration Register Group Dialog Box.....	5-1
5.1.2	Local Configuration Register Group Dialog Box	5-3
5.1.3	The Runtime Register Group Dialog Box.....	5-7
5.1.4	The DMA Register Group Dialog Box	5-8
5.1.5	The Messaging FIFO Register Group Dialog Box.....	5-10
6.	The PCI 9080 Register Set.....	6-1
6.1	The Register Group Dialog Boxes	6-1
6.1.1	PCI Configuration Register Group Dialog Box.....	6-1
6.1.2	Local Configuration Register Group Dialog Box	6-2
6.1.3	The Runtime Register Group Dialog Box.....	6-6
6.1.4	The DMA Register Group Dialog Box	6-7
6.1.5	The Messaging FIFO Register Group Dialog Box.....	6-9
Appendix A.	Troubleshooting.....	A-1
Appendix B.	Glossary Of Terms	B-1

LIST OF FIGURES

FIGURE 2-1	THE PLXMON 99 INTERFACE	2-1
FIGURE 2-2.	SELECTING THE PROPERTIES	2-1
FIGURE 2-3.	DEVICE CONFIGURATION	2-2
FIGURE 2-4.	EEPROM EDIT UTILITY	2-3
FIGURE 2-5.	DOWNLOAD TO IOP.....	2-4
FIGURE 2-6.	PCI 9080 AND 9054 DMA CHANNEL 0 REGISTERS	2-6
FIGURE 2-7.	IOP 480 DMA CHANNEL 0 REGISTERS	2-6
FIGURE 2-8.	PLXMON 99 STARTUP SCREEN	2-7
FIGURE 2-9.	PCI 9054RDK-LITE BOARD PROPERTIES	2-8
FIGURE 2-10.	PROPERTIES FOR BOARD WITH NO CPU.....	2-8
FIGURE 2-12.	BOARD WITH NO CPU PROPERTIES.....	2-1
FIGURE 3-1.	PCI VS. SERIAL DATA FLOW.....	3-1
FIGURE 3-2.	APPLICATION HOT LINKS DIALOG BOX	3-2
FIGURE 3-3.	DEVICE CONFIGURATION DIALOG BOX.....	3-2
FIGURE 3-4.	FILE DOWNLOAD DIALOG BOX	3-3
FIGURE 3-5.	FONT SELECT DIALOG BOX.....	3-4
FIGURE 3-6.	THE PLXMON 99 INTERFACE	3-4
FIGURE 3-7.	COMMAND LINE INTERFACE COMMANDS	3-5
FIGURE 3-8.	INTERACTIVE MODE.....	3-6
FIGURE 3-9.	USER VARIABLES AND THEIR DEFINITIONS.....	3-7
FIGURE 3-10.	DEVICE SELECT DIALOG BOX	3-8
FIGURE 3-11.	SERIAL COMMUNICATIONS PROPERTIES DIALOG BOX	3-9
FIGURE 3-12.	SERIAL EEPROM ACCESS SCREEN	3-9
FIGURE 4-1.	PCI CONFIGURATION REGISTERS DIALOG BOX	4-1
FIGURE 4-2.	LOCAL CONFIGURATION REGISTERS DIALOG BOX.....	4-2
FIGURE 4-3.	RUNTIME REGISTER GROUP DIALOG BOX.....	4-2

FIGURE 4-4. INTERRUPT ENABLE /STATUS REGISTER DIALOG BOX.....	4-3
FIGURE 4-5. DMA REGISTER GROUP DIALOG BOX	4-3
FIGURE 4-6. MESSAGING QUEUE REGISTER GROUP.....	4-4
FIGURE 4-7. IOP 480 MEMORY CONTROLLER REGISTERS.....	4-5
FIGURE 4-8. IOP 480 CPU REGISTERS GROUP	4-6
FIGURE 4-9. IOP 480 EEPROM VALUES.....	4-7
FIGURE 5-1. PCI CONFIGURATION REGISTERS DIALOG BOX	5-1
FIGURE 5-2. POWER MANAGEMENT CAPABILITIES DIALOG BOX	5-2
FIGURE 5-3. POWER MANAGEMENT CSR DIALOG BOX	5-2
FIGURE 5-4. HOT SWAP CSR DIALOG BOX	5-2
FIGURE 5-5. LOCAL CONFIGURATION REGISTERS DIALOG BOX	5-3
FIGURE 5-6. MODE/ARBITRATION DIALOG BOX.....	5-3
FIGURE 5-7. ENDIAN DESCRIPTOR DIALOG BOX.....	5-4
FIGURE 5-8. LOCAL MISCELLANEOUS CONTROL REGISTER DIALOG BOX	5-4
FIGURE 5-9. REGION 0/ROM DESCRIPTOR DIALOG BOX.....	5-5
FIGURE 5-10. DIRECT MASTER REMAP DIALOG BOX	5-5
FIGURE 5-11. DIRECT MASTER CONFIG. DIALOG BOX.....	5-6
FIGURE 5-12. REGION 1 DESCRIPTOR DIALOG BOX.....	5-6
FIGURE 5-13. RUN TIME REGISTERS DIALOG BOX	5-7
FIGURE 5-14. INTERRUPT CONTROL AND STATUS DIALOG BOX	5-7
FIGURE 5-15. EEPROM PCI USER IO DIALOG BOX	5-8
FIGURE 5-16. LOCAL DMA REGISTERS DIALOG BOX	5-8
FIGURE 5-17. DMA MODE DIALOG BOX.....	5-9
FIGURE 5-18. DMA DESCRIPTOR POINTER DIALOG BOX	5-9
FIGURE 5-19. DMA THRESHOLDS DIALOG BOX.....	5-9
FIGURE 5-20. MESSAGING UNIT REGISTERS DIALOG BOX.....	5-10
FIGURE 5-21. STATUS/CONTROL REGISTER DIALOG BOX	5-10
FIGURE 6-1. PCI CONFIGURATION REGISTERS DIALOG BOX.....	6-1
FIGURE 6-2. LOCAL CONFIGURATION REGISTERS DIALOG BOX	6-2
FIGURE 6-3. DIRECT MASTER MODE / ARBITRATION DIALOG BOX	6-3
FIGURE 6-4. ENDIAN DESCRIPTOR DIALOG BOX.....	6-3
FIGURE 6-5. LOCAL SPACE 0/EMP ROM DIALOG BOX.....	6-4
FIGURE 6-6. DIRECT MASTER PCI REMAP DIALOG BOX	6-4
FIGURE 6-7. DIRECT MASTER CONFIGURE DIALOG BOX	6-5
FIGURE 6-8. REGION 1 DESCRIPTOR DIALOG BOX.....	6-5
FIGURE 6-9. RUNTIME REGISTERS DIALOG BOX.....	6-6
FIGURE 6-10. INTERRUPT CONTROL AND STATUS DIALOG BOX	6-6
FIGURE 6-11. EEPROM, PCI, USER IO DETAILS DIALOG BOX	6-7
FIGURE 6-12. DMA REGISTERS DIALOG BOX	6-7
FIGURE 6-13. DMA MODE DIALOG BOX.....	6-8
FIGURE 6-14. DESCRIPTOR POINTER DIALOG BOX	6-8
FIGURE 6-15. DMA CHANNELS THRESHOLD DIALOG BOX	6-8
FIGURE 6-16. MESSAGING UNIT REGISTERS DIALOG BOX.....	6-9
FIGURE 6-17. FIFO STATUS/CONTROL REGISTER DIALOG BOX.....	6-9

LIST OF TABLES

TABLE 2-1. RDK BOARDS SUPPORTED BY PLXMON 99	2-3
--	-----



This page intentionally left blank.

1. Introduction

PLXMon 99 program is a powerful Windows based GUI debug utility that allows easy configuration, viewing, and modification of various registers on PLX's PCI devices. It also allows the user to perform interactive block-mode, scatter-gather, and shuttle mode DMA operations. PLXMon 99 incorporates a built-in downloader application for downloading application specific code to your target hardware and supports both FLASH and EEPROM programming. PLXMon 99 works with Windows 98 and Windows NT 4.0 operating systems. It is designed to operate with PLX's Reference Design Boards and customer specific hardware that contains either the IOP 480, PCI 9054, or PCI 9080 devices. It includes an EEPROM Edit Utility to allow programming of blank EEPROMs.

The dialog box windows contain detailed information about each register and if necessary about the individual bits within. Combining this information with the programming algorithms included for accessing the various IOP components and the ability to communicate over both PCI and Serial buses, PLXMon 99 gives the tools needed to access the PLX RDKs and your prototype board properly.

PLXMon 99 is automatically installed on your system when you install PCI SDK v3.0. It is compatible with PLX's IOP 480, PCI 9054 and PCI 9080 devices only. For PLX PCI 9054RDK-Lite board, PLXMon 99 is included on the PCI 9054RDK_Lite HDK CD-ROM.

1.1 About This Manual

This manual is divided into five chapters. Chapter one is the Introduction, chapter two shows you how to set up PLXMon 99 quickly, and see how some of the more common functions are used. Chapter three gives a self-guided tour of PLXMon 99. In this area every feature found in PLXMon 99 is outlined. Chapter 4, 5, and 6 describe the PLXMon 99 GUI screens for IOP 480, PCI 9054, and PCI 9080 devices respectively. Appendix D is the troubleshooting section and Appendix E is the glossary of terms.

1.2 Conventions And Support

References to Windows NT assume Windows NT 4.0 or higher and will be shown as WinNT. Similarly, references to Windows98 will be shown as Win98.

All references to IOP (I/O Platform) throughout this manual refer to the embedded hardware and all references to IOP software refer to the embedded software.

All values used in the manual are hexadecimal numbers, with the exception for memory sizes (used in Local Configuration Register screen). The prefix '0x' has been omitted from all hexadecimal numbers and is not required when entering values for PLXMon 99 fields.

It is important to note that PLXMon 99 can only operate with a PLX device. This version has been designed to work with the following PLX's hardware platforms:

- IOP 480RDK
- PCI 9054RDK-Lite
- PCI 9054RDK-860
- CompactPCI 9054RDK-860
- PCI 9080RDK-401B



- PCI 9080RDK-RC32364
- PCI 9080RDK-SH3
- PCI 9080RDK-860
- Any customer board that uses the IOP 480, PCI 9080 or PCI 9054 chips.

PLXMon 99 has been tested to ensure compatibility with both Windows NT and Windows98.

Note: PLXMon 99 is designed to run best with a high-resolution display, such as 1024x768 with 256 colors or better. Users choosing to run the software at lower resolutions will find it visually undesirable.

1.3 Installation

In most cases, PLXMon 99 program is installed on your system when you install PCI SDK v3.0. It supports Windows 98 and Windows NT 4.0 operating systems. You can launch this program by clicking on the “P-PLXMon 99” icon in the PCI SDK folder in the Start Menu. For PCI 9054RDK-Lite board, PLXMon 99 can be installed from the CD-ROM included in the kit.

1.4 PLXMon 99 Feature List

PLXMon 99 provides the following features:

- Graphical User Interface (GUI) screens that are based on PLX device registers
- Compatible with the IOP 480, PCI 9080 and PCI 9054 chips
- EEPROM Edit Utility to program a blank EEPROM
- Split Screen Interface, allowing command line input while receiving serial data.
- Serial communications with an IOP’s debug port. This feature is compatible with PLX’s Back-End Monitor protocol
- A built-in downloader providing support for the following image standards: Motorola S-Record, IBM-401B Image Files, COFF, and Binary. This feature supports downloading to RAM and FLASH devices through the PCI bus and the serial port.
- PLX EEPROM Configuration screens to modify the contents of NM93CS46, NM93CS56, and NM93CS66 EEPROMs. These configuration screens allow you to load and save values from and to a file.
- Customizable Hot-Links. This feature allows users to launch Win32 compatible programs such as testing and sample programs

Note: PLXMon 99 is only compatible with PLX IOP 480, PCI 9054, and PCI 9080.



1.5 Customer Support Information

Prior to contacting PLX customer support, please ensure that you are situated close to the computer that has the PCI SDK installed and have the following information:

1. Model number of the PLX PCI RDK (if any)
2. PLX PCI SDK version (if any)
3. Host Operating System and version
4. PLXMon 99 version
5. Description of your intended design:
 - PLX chip used
 - Microprocessor (if any)
 - Local Operating System and version (if any)
 - I/O devices (if any)
6. Description of your problem
7. Steps to recreate the problem

You may contact PLX customer support at:

Address: PLX Technology, Inc.
Attn. Technical Support
390 Potrero Avenue
Sunnyvale, CA 94086

Phone: 408-774-9060
Fax: 408-774-2169
Web: <http://www.plxtech.com>

You may send email to one of the following addresses:

west-apps@plxtech.com
mid-apps@plxtech.com
east-apps@plxtech.com
euro-apps@plxtech.com
asia-apps@plxtech.com



This page intentionally left blank.

2. A Brief Tour of PLXMon 99

This section will give a 10-minute tour of the PLXMon 99 program. During this tour you will become familiar with how to setup the program. Some of the more common features found in PLXMon 99 will also be discussed.

2.1 Starting PLXMon 99 And Configuration

PLXMon 99 can be started by:

- Clicking on the icon in the PCI SDK folder in the Start Menu.

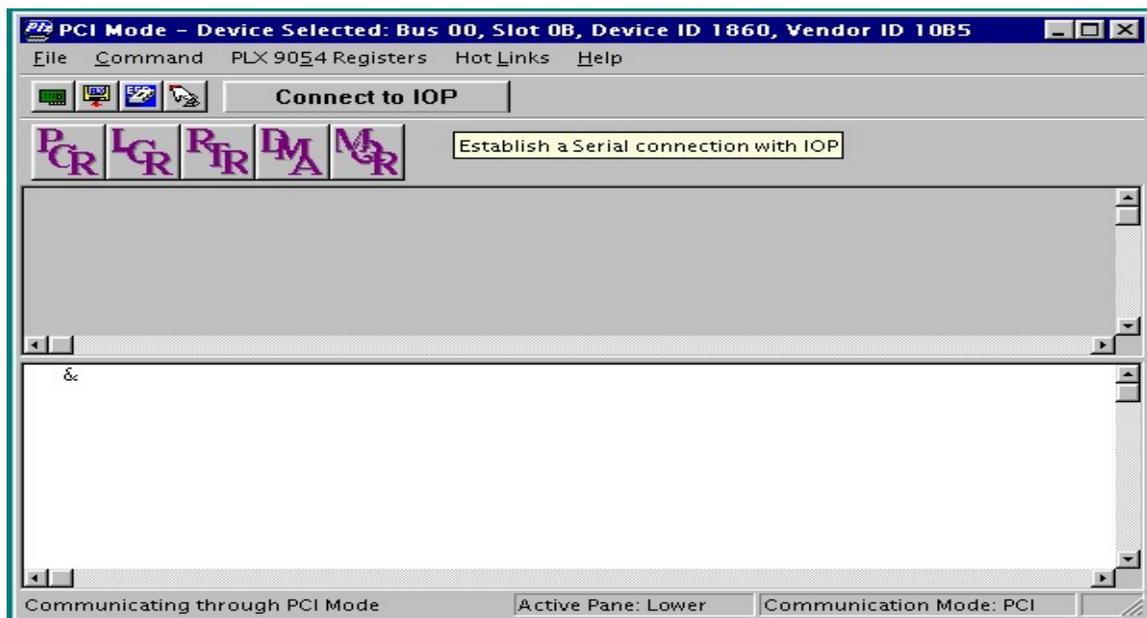


Figure 2-1 The PLXMon 99 Interface

Depending on whether a supported PLX RDK is present, the monitor will enter PCI Mode or Local IOP (Serial) Mode. If a PLX RDK is in a PCI slot on your computer you will be in PCI mode, otherwise you will enter Local IOP (Serial) Mode. It is possible to switch back and forth between the two modes after the program has started. Also note that this program can be run on a second computer and may be used to do remote debugging through a serial port. PLXMon 99 recognizes boards by checking the vendor and device IDs.

If PLXMon 99 does not detect a PLX RDK in the system, it will notify the user that one does not exist. If you are using your own board, you must add a new device to the Properties menu or verify that the device data is correct in the

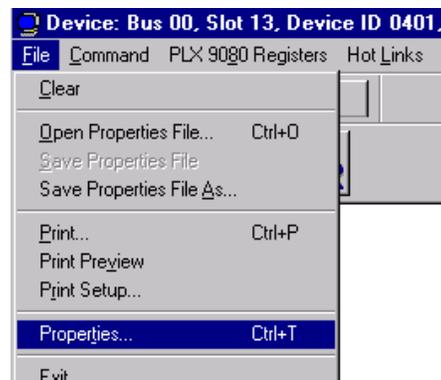


Figure 2-2. Selecting the Properties



Properties menu. Until properties are assigned to the specific Vendor and Device ID of your board, the program will not know the data needed to access the board. To edit these values select the Properties item under the File pull-down menu. (You can also use the hot-key Ctrl-t).

The Properties menu first shows the Device Configuration page. Figure 2-3 shows the Device Configuration screen for IOP 480RDK. This data is used for PCI access only, the serial configuration data is retrieved using Back End Monitor commands directly through the serial port. The Properties data will be correct if you are using a PLX RDK board. For IOP 480RDK board, under Local CPU Options, Big Endian CPU is selected by default because IOP 480 supports Big Endian.

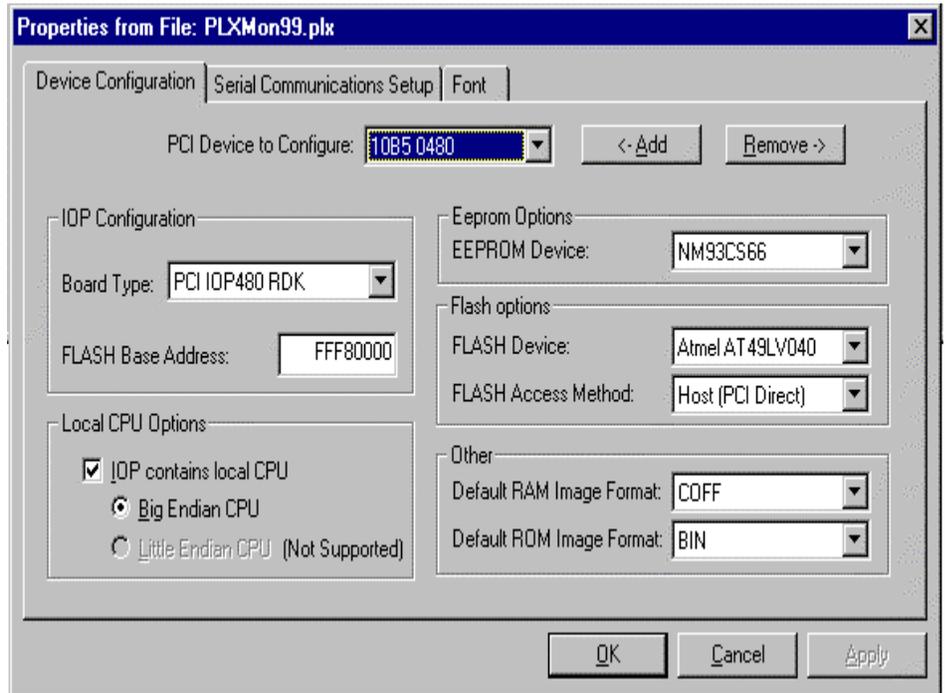


Figure 2-3. Device Configuration

These are the steps that should be followed to ensure proper operation of PLXMon 99:

1. If the Vendor and Device ID are not listed under the “PCI Device to Configure” combo box, then click the “ADD” push button to create a new custom property entry for this device.
2. Now enter all the valid device information. By first clicking the RDK Default button, the default values for the RDK type selected will be entered on the screen. You should still verify that the Configuration EEPROM and FLASH EEPROM are the correct models.
3. Change the memory map, or set default data file extension types, if necessary.

Now click the Serial Communications tab.



If you wish to use the debug port on your RDK, then you need to select the correct COM port. Also note if you are in Serial mode (entered by clicking on Connect to IOP) when doing this change, you must switch out then back into Serial mode for your changes to take effect. The IOP software on the PLX RDKs has been designed to accept 38400 Baud only. You can also change the timeout length if desired. To see command line data in a different font, select the font tab to select many different styles and point sizes. Select the underlined option and it will turn the display screen into lined pages.

2.1.1 Default Vendor and Device IDs for PLX Reference Design Boards

PLXMon 99 supports a number of PLX Reference Design Boards. They are listed in Table 2-1 along with their vendor and device IDs for your reference.

Table 2-1. RDK Boards Supported by PLXMon 99

Name of PLX Boards	Default Device ID	Default Vendor ID
IOP 480RDK	0480	10B5
PCI 9054RDK-860	1860	10B5
CompactPCI 9054RDK-860	C860	10B5
PCI 9054RDK-Lite	5406	10B5
PCI 9080RDK-860	0860	10B5
PCI 9080RDK-401B	0401	10B5
PCI 9080RDK-SH3	7709	10B5
PCI 9080RDK-RC32364	0364	10B5

2.2 Displaying Registers

Whether you are in PCI mode or Serial mode, you can access the registers on a PLX RDK. It's as easy as clicking a button. The large buttons on the lower tier display various register sets. Click on the PCI Configuration Registers (PCR) button. A formatted PCI register set appears with some bit decoding already done. Grayed-out boxes indicate read-only windows. Now close this window and open the Local Configuration Register (LCR) window.



By clicking on an edit box, you can change the hexadecimal value. Then either close the window or move the cursor to another edit box to enter the value. Clicking on a check box will automatically update the register. Try it by clicking on the Details of any register box. The next dialog that appears will have check boxes to represent various bits of the register.

All the register capabilities mentioned above are applicable in Serial Mode as well as in PCI mode. PLXMon 99 uses the interface provided by the Back End Monitor (BEM) to read and write to register locations. For more information on BEM, see either the PCI SDK Programmers Manual or the glossary.

2.3 Popular Features

Now you can try using the built in applications that make PLXMon 99 very useful. This section will give you a brief overview of the interactive downloader application as well as interactive DMA.

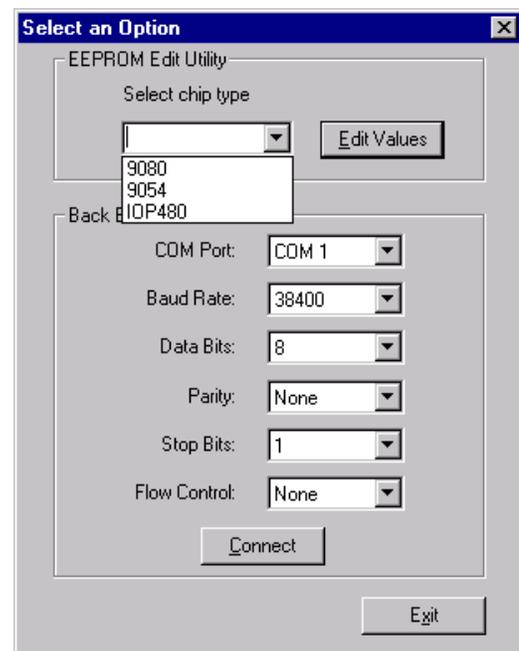


Figure 2-4. EEPROM Edit Utility

2.3.1 EEPROM Edit Utility

The EEPROM Edit Utility allows you to program a blank EEPROM. (Figure 2-4)

Note: This screen is only available when you start PLXMon 99 with no board plugged into your PC.

You simply select the PLX chip you are using and click on <Edit Values> push button. This will take you to the chip specific EEPROM screen. All values on this screen are zero. On this screen, you can either enter your own values or load the values from a file. (The PCI SDK contains a *.eep file for each PLX board in the ...\\hw\\eeprom\\... directory). Then, you can save this file to a floppy disk and walk over to a I/O programmer and program your blank EEPROM device.

2.3.2 Downloading To The IOP

If you want to download an application to the IOP side, this is the feature you will need to use. Some of the features of the download utility include:

- Translation from different file formats including COFF, IBM ELF, Motorola SRecord, and pure Binary.
- The ability to download the file to either RAM or FLASH ROM.
- Binary reads from FLASH ROM to a binary data file.
- Supports Serial downloads to RAM.

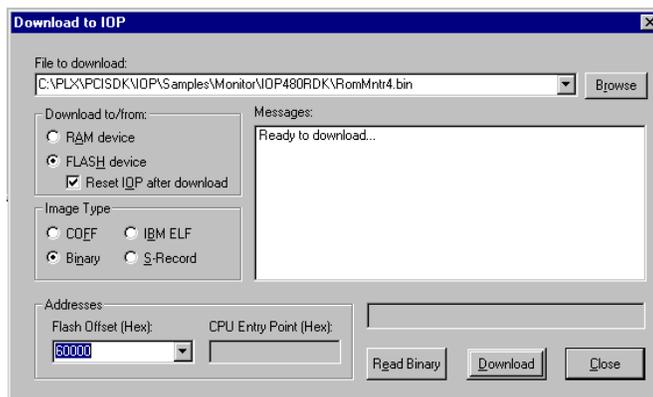


Figure 2-5. Download To IOP

To familiarize yourself with downloading an application/image to an RDK or to your target board, this tour will take you through the steps of downloading a RAM Hello Sample, and the Direct Master ROM sample, both of which are found in PCI SDK v3.0. The examples below demonstrate downloading using the PCI 9054RDK-860 but the methods also apply to other PLX RDKs.

Downloading to RAM

- a. You must first select either a Serial or PCI channel before opening the Download window. If performing a PCI download you can connect a serial port to another PC running PLXMon 99 application, and see the download in progress.

- b. Open the Download Window by clicking the download to embedded icon, which looks like a small disk with a  downward facing arrow.
- c. In the Device Configuration information, RAM data is stored in COFF format for the currently selected device. Therefore the default file type will be COFF.
 1. Go to the directory:
 <Install-Path>\PLX\PciSdk300\IOP\Samples\Hello\9054RDK-860 and select the file RamHello.cof.
 2. Click on download. You can verify that the program was successfully downloaded by reading the status screen (See Figure 2-4).

Burning a FLASH ROM

Note: If in the course of writing a program to the FLASH an error occurs, do not shut down the computer until you have re-burned a valid ROM program. Failure to do so will require removal of the FLASH ROM chip and re-burning in an external device programmer.

1. Downloads to FLASH are supported via the PCI bus and Serial port. In this example, we will be downloading an image to the FLASH via the PCI bus. So, select the FLASH device.

2. After selecting FLASH, set the desired image type. It will also be necessary to give an offset from the physical FLASH address set in the configuration menu. This is necessary because certain RDKs produce FLASH data to be downloaded to different addresses. The following chart describes the offsets used for each RDK.

RDK Type	FLASH OFFSET (in HEX)
IOP 480RDK	60000
PCI 9054RDK-860	0
CompactPCI 9054RDK-860	0
PCI 9080RDK-860	0
PCI 9080RDK-401B	60000
PCI 9080RDK-RC32364	N/A
PCI 9080RDK-SH3	0
PCI 9054RDK-Lite	0

3. Go to the directory:
 <Install-Path>\PLX\PciSdk300\IOP\Samples\Dmaster\9054RDK-860 and select the file RomDM.bin.
4. Now click on download. Again you can verify that the download was successful by reading from the serial port on another PLXMon 99, and by reading the Status window as well. A good way to test the ROM code is to reset the RDK, and verify the same program is run after the reset.

For a more complete description of the IOP download function consult the reference section.

2.3.3 DMA Transfers

DMA transfers can be used to transfer data rapidly between the IOP and the PCI bus. This PCI bus could be a user-mapped common buffer, or another RDK's local space window. This example will demonstrate how to set up a simple IOP to PCI DMA transfer.

1. To set up the DMA transfer, some registers must be properly initialized. First click on the



DMA button, to display the registers to be modified. Set up the DMA transfer as shown in Figure 2-6.

2. At the & prompt, which is located in the Lower Pane, various commands can be entered to get information on the RDK board that is selected. The “VARS” command (started by typing “vars” at the & prompt) gives two values for Hbuf, a virtual and a physical address. The physical address is used for the PCI Address (Lower 32 bits). Enter a value for Transfer Count, 500 is used in this example. Entering dl s0+100000 shows you the current contents of the local memory on PLX RDK board at location 1MB. You can use the el command for editing and writing to local memory.
3. The typical requirement for setting up the mode register is to set the Ready Input Enable and the bus width to 32 bit for PCI 9054 and PCI 9080 based RDKs(see figure 2-6). These two combinations give the Mode register a value of 43(h). For the IOP 480RDK, the Mode register value can be 0(see figure 2-7). The Local Address given here is valid for all supported RDKs and is 100000(h). The Data Transfer direction is set within the Descriptor Pointer by clicking on Details button, and setting the direction to Local to PCI. The Threshold Register has a default value of 0.

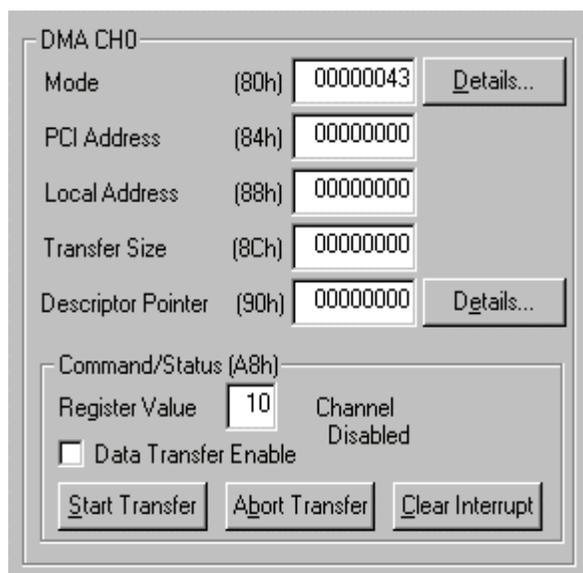


Figure 2-6. PCI 9080 and 9054 DMA Channel 0 Registers

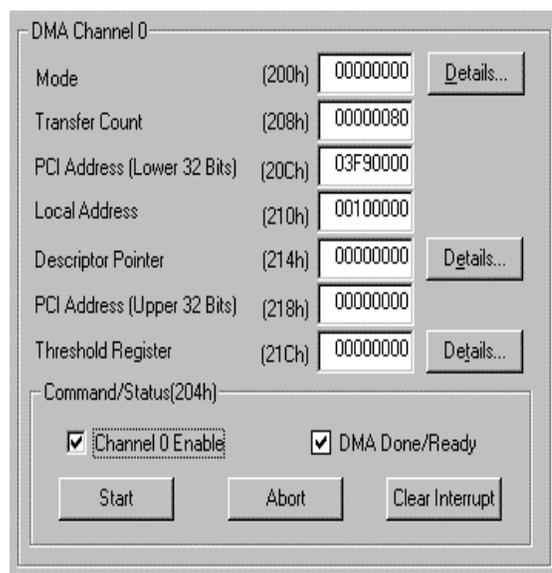


Figure 2-7. IOP 480 DMA Channel 0 Registers

4. To start the transfer, first enable the transfer by checking the Channel 0 Enable box. Clicking on Start button will begin the operation. The DMA Done/Ready bit will remain checked because the transfer count of 80 is very small and the transfer completes very quickly.
5. To verify the data was transferred correctly, you can read the PCI buffer, Hbuf by typing dl hbuf at the & prompt in the lower pane.

This concludes the tour of PLXMon 99. More details about the specific PLX devices are found in the appendices.

2.4 Board with No Local CPU

If you are using the PCI 9054RDK-Lite board and/or if you are not using a local CPU on your board, then the following information may be helpful to you.

2.4.1 Using PCI 9054RDK-Lite Board or Your Custom Board with No CPU

The PCI 9054RDK-Lite box contains a Hardware Development Kit CD-ROM. This CD-ROM contains PLXMon 99 - a Windows based debug utility, a Windows NT and a Windows 98 driver. You can install PLXMon 99 on a Windows NT or a Windows 98 PC by inserting this CD-ROM in your PC's CD-ROM drive. If you are using your custom board with no local CPU, you should first install the PCI SDK v3.0 (available from PLX, software@plxtech.com).

In both of the above cases, when you start PLXMon 99, you will get the screen shown in Figure 2-8 below:

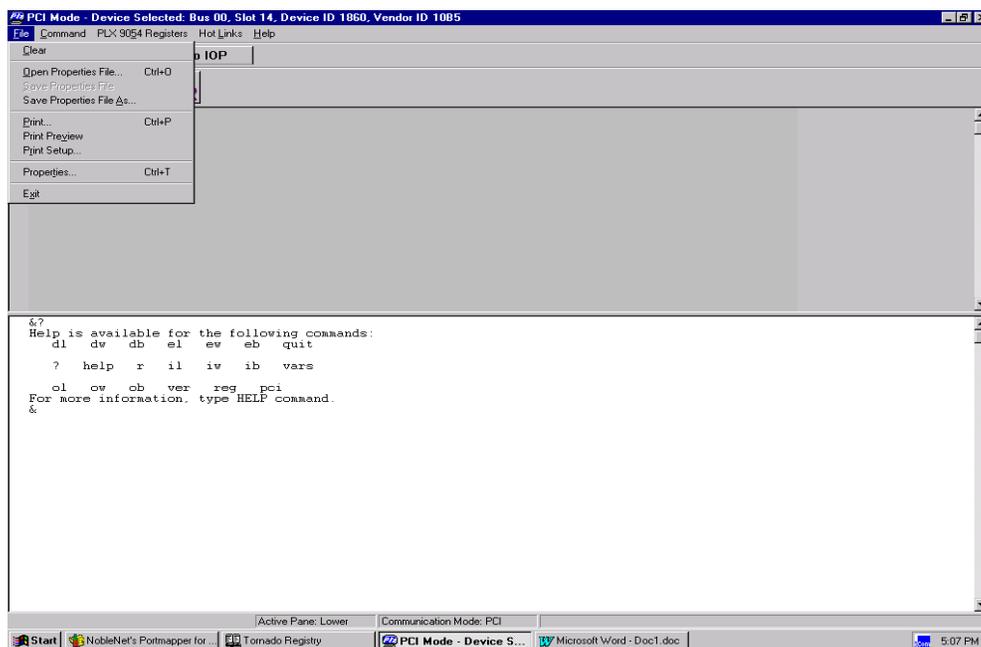


Figure 2-8. PLXMon 99 Startup Screen



From the <File> pull down menu, click on <Properties> and go to the <Device Configuration page>.

If you are using the PLX PCI 9054RDK-Lite board, then from the <Board Type> pull down menu, you should select <PCI 9054RDK-LITE>. Figure 2-9 shows the settings for the PCI 9054RDK-Lite board.

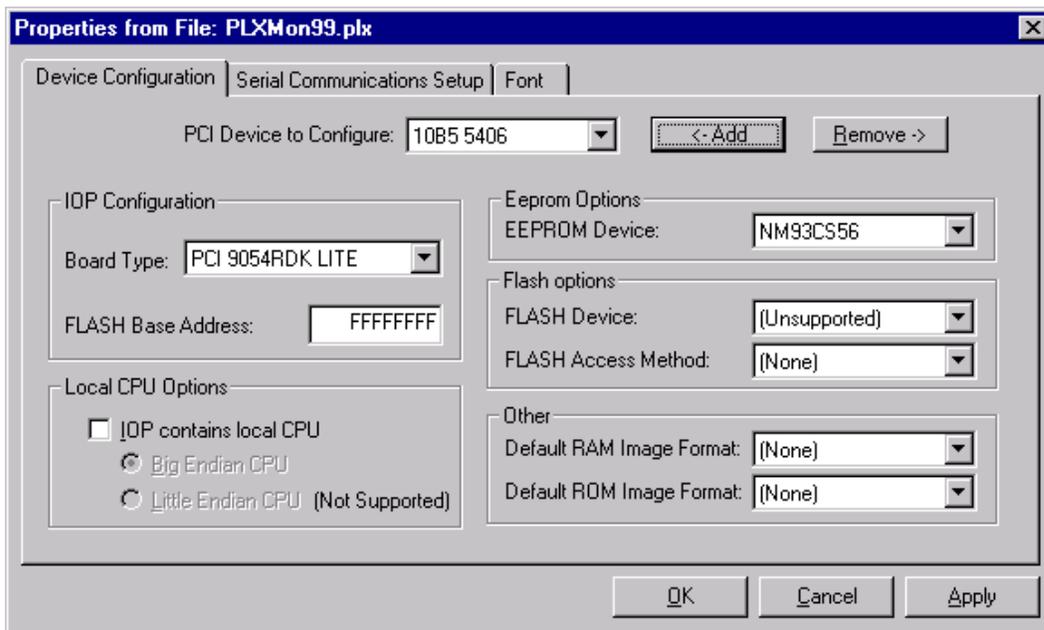


Figure 2-9. PCI 9054RDK-Lite Board Properties

If your board does not have a local CPU, you should de-select the <IOP contains local CPU> check box under the <Local CPU Options>. Figure 2-10 shows the Device Configuration screen for a custom board with a Device Id of ABCD and a Vendor Id of 10B5. These Ids were added by clicking on the <-Add> push down button. Next, you should configure all or some of the other options on this screen depending upon your board design.

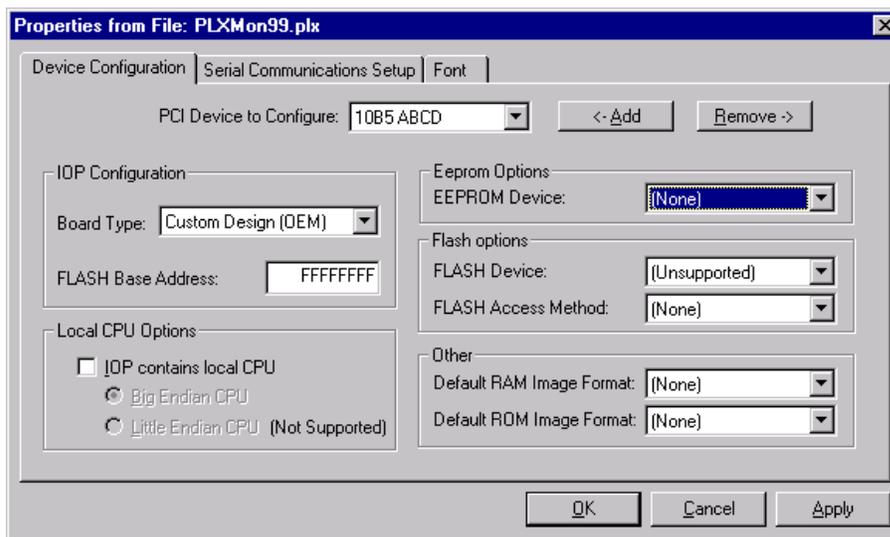


Figure 2-10. Properties for Board with No CPU

3. PLXMon 99 - Reference

The reference section is provided to give detailed information about every feature of the PLXMon 99 application. It is organized alphabetically by feature title.

3.1 Access Mode

The two types of access to the RDKs are via the PCI bus and via the Serial port. See Figure 3-1.

3.1.1 PCI Mode

When PCI mode is selected all communication between PLXMon 99 and the PLX chip is done via the PCI SDK PCI (Host) API and Windows device driver. This method will be familiar to users of PLXMon 97 and PLXMon 98. The mode is selected by toggling the Connect to IOP/Disconnect from IOP button on the toolbar. In PCI mode, Connect to IOP button is displayed. Essentially, the PCI mode results in all communication to the PLX device via its PCI Bus interface. All RDK properties used in PCI communication are found in the Device Configuration menu. Figure 3-1 shows PLXMon 99 interface in PCI mode.

3.1.2 IOP (Serial) Mode

IOP Mode is entered by clicking on “Connect to IOP” toggle button. PLXMon 99 automatically enters IOP mode if a PLX RDK is not present or if you are using your prototype board for the first time. When IOP (serial communication) mode is selected all communication between PLXMon 99 and the PLX device by passes the PCI SDK API and device driver. Instead, PLXMon 99 communicates with the IOP’s BEM module (refer to PCI SDK Programmer’s Manual for BEM details). First the program queries RDK information such as the PLX register base address. Essentially this results in all communication to the PLX device via its Local Bus interface. Commands are limited to local reads and writes, and a local reset.

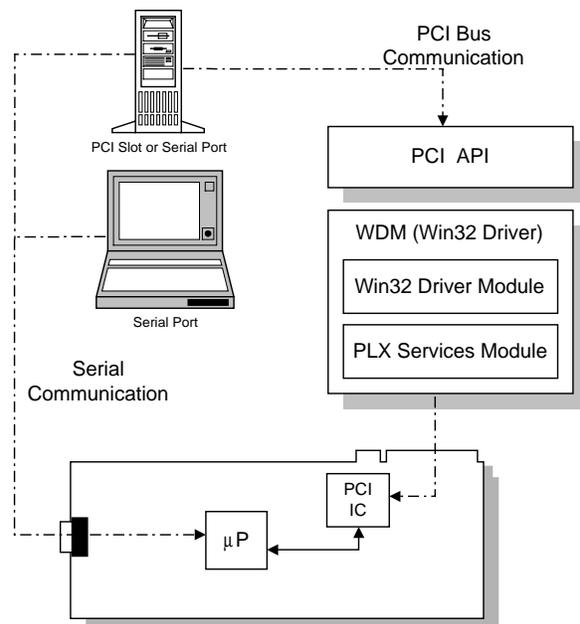


Figure 3-1. PCI vs. Serial Data Flow

Note: To use this mode, your IOP software must contain the BEM module. All PLX RDKs support this mode by default.

3.2 Application Hot Links

The application hot links can be started by selecting Hot Link from the PLXMon 99 pull down menu bar. The PLXMon 99 offers the capability for users to add hot links to popular SDK applications. Typical applications that users may like to add hot links to are PCI SDK samples, PLX manufacturing test software, or custom applications. Please see Figure 3-2.

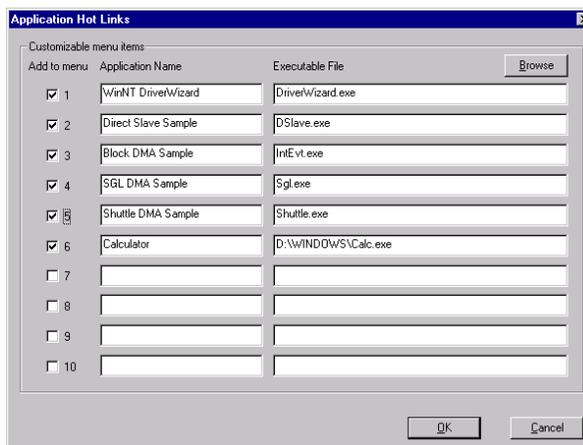


Figure 3-2. Application Hot Links Dialog Box

To use this feature you need to enter the Application Name and associated path to the executable in the Hotlinks pull-down menu. They will then appear in the Hot Link pull down menu.

3.3 Device Configuration

This dialog window can be found by pulling down the File menu then selecting Properties. PLXMon 99 remembers program options for specific PLX RDKs and custom boards. These options are used throughout PLXMon 99 to setup default values for various program options. For example the flash properties are used by the loader to determine the flash programming method needed.

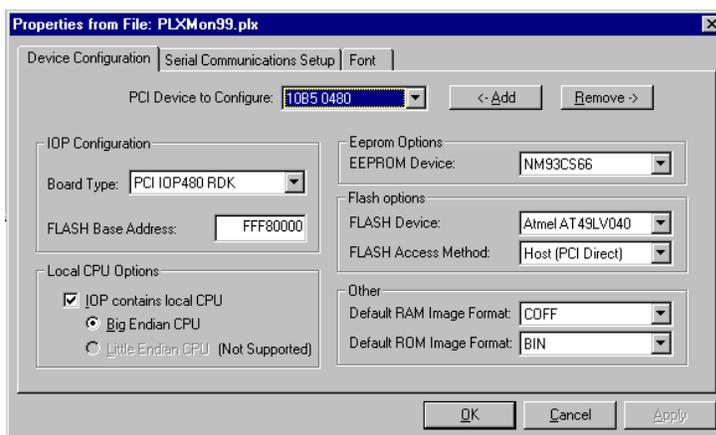


Figure 3-3. Device Configuration Dialog Box

All supported device attributes are saved to a **Properties File**. This file has a default name PLXMon99.plx and is automatically loaded. PLXMon 99 will only search the current directory for this file so if the executable is moved, this file should be moved as well.

PLXMon 99 uses the Vendor ID and Device ID to recognize a **supported** device. By adding a new entry to this list, a custom ID combination can be supported. Figure 3-3 shows the settings for the IOP 480RDK board. Similar settings must be selected for each RDK that PLXMon 99 is used with.

When adding a new device, first enter the custom Vendor ID and Device ID. Then select its RDK type (if applicable). On hitting RDK Default, the normal settings for that RDK will be entered. Press Apply or OK to enter the data into the Properties file.

3.4 Downloading IOP Applications

In order to download to and execute programs on the IOP, the Download to IOP command is provided. This utility can be run by clicking on the icon, or by selecting it under the Command pull-down menu.

Files can be sent to either a RAM device or FLASH device and the base address is programmable.

Typically, all default values that are already selected will be correct for the download you wish to do. By selecting either RAM or FLASH, the file format will automatically be changed to the

format that is specified for this device in the Device Config. menu. These features can be overridden, and by doing so the user should be knowledgeable about the format of these files and the memory map of the RDK being programmed.

***Note:** Programming the FLASH can be dangerous and it is important that the setup parameters be correct before the download is attempted. Data like RDK Type, Flash Address, Programming Method, and Memory Offset must be known beforehand.*

To read the data on a FLASH device into a file, the Read Binary button can be used in conjunction with the memory offset window. This data will be retrieved unformatted and stored as a pure binary file. Currently the binary read function will create an image that starts at the memory offset provided and ends at the end of the usable flash range.

This utility also has the ability to program a device through the serial debug port. The serial download supports both ROM and RAM programming.

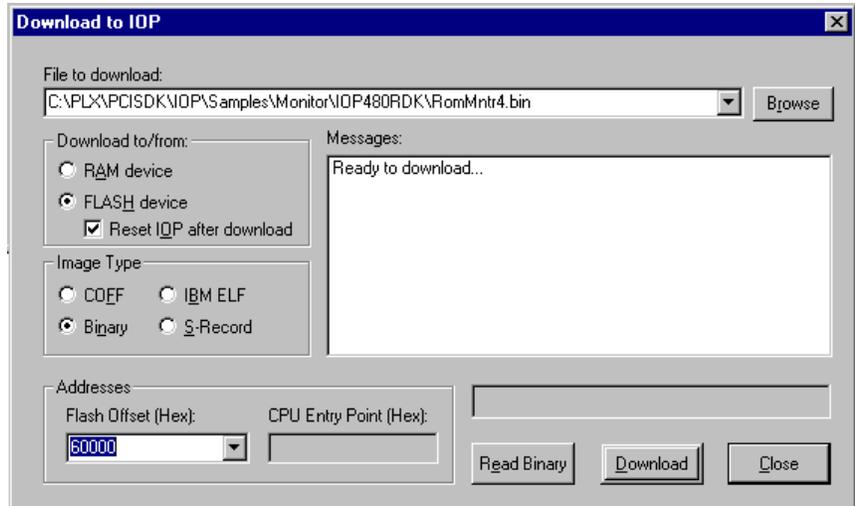


Figure 3-4. File Download Dialog Box

3.5 Font Configuration

This dialog screen can be reached from the Properties option in the File pull-down menu. The font and point size will be changed in the display screen only. The appearance of the data in the dialog boxes will not be changed.

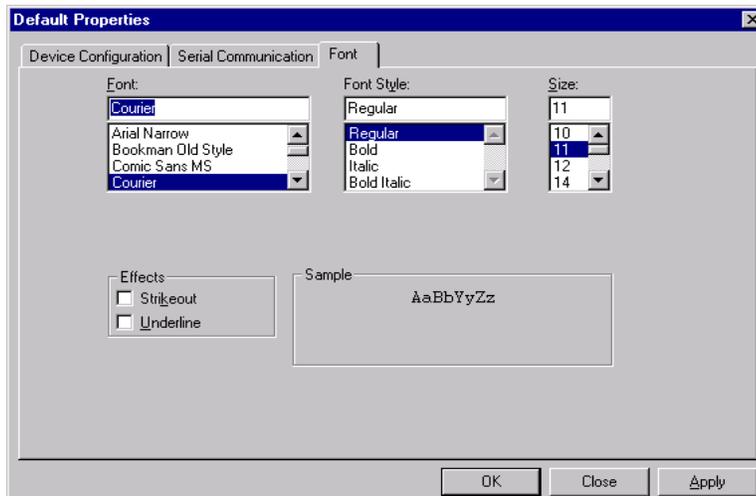


Figure 3-5. Font Select Dialog Box

3.6 The Interface

The PLXMon 99's main interface, shown in Figure 3-6 contains:

- A drop-down menu bar, with the five main drop-down menus. Depending on which PLX RDK is selected; certain options will be available. The PLXMon 99 interface for 9054RDK menu is shown in Figure 3-6.
- A split screen interface used for simultaneous Command Line Interface (Lower Pane) and Serial Access (Upper Pane). Use the F6 key to toggle between active panes.

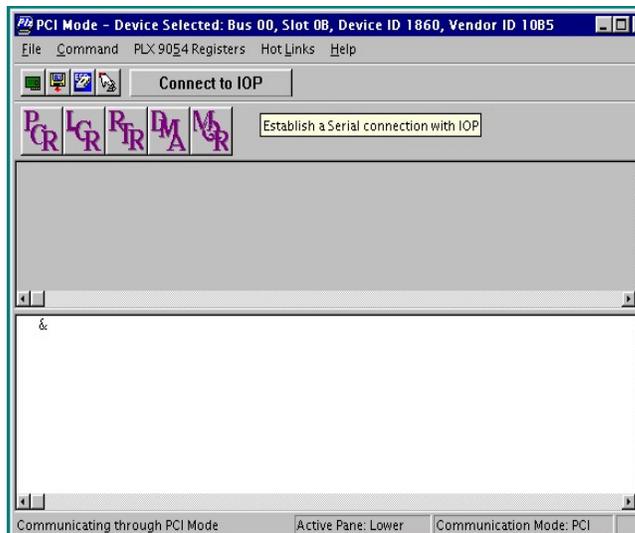


Figure 3-6. The PLXMon 99 Interface

- An optional toolbar
- The status bar which reports the configuration file being used
- The Connect to IOP/Disconnect from IOP button toggles between PCI and serial communication mode.

3.6.1 The PLXMon 99 Toolbar

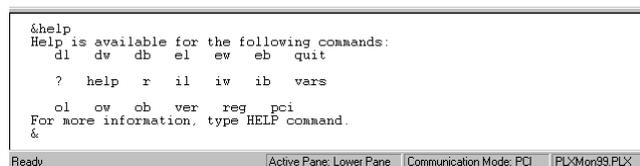
The PLXMon 99 toolbar serves as an optional shortcut to the drop-down menu commands.

3.6.2 Status Bar

The status bar provides simple and useful tips. When the mouse is pointing on an object or a button, in the main window of PLXMon 99 the status bar displays information about it. Tool-tips are also displayed when mouse pointer is held over an object for a short period.

3.6.3 Command Line Interface (CLI)

At the `&` prompt which is located in the Lower Pane, various commands can be entered to get information on the RDK that is selected. This command line can be used in both PCI and Serial modes. To get a list of valid commands at any time in



```

&help
Help is available for the following commands:
dl  dw  db  el  ew  eb  quit
?   help r  il  iw  ib  vars
ol  ow  ob  ver reg pci
For more information, type HELP command.
&

```

PLXMon 99, at the command line, type `help` or `?`. The following sections will describe all the valid commands.

Figure 3-7. Command Line Interface Commands

Note: The CLI is not case sensitive.

3.6.3.1 Displaying Memory Via Memory Cycles, (dl, dw, db)

These commands display different sizes of data that are accessed through memory cycles. Using `dl` will return a 32-bit value, `dw` will return a 16-bit value, and `db` will return a 8-bit value.

They follow the format:

```
<command> <address> [[1] bytelength]
```

By default the bytelength is 80 [hex] bytes. Typing the command again with no arguments will make PLXMon 99 continue to display the range with the same bytelength as before.

3.6.3.2 Displaying Memory Via I/O Cycles, (il, iw, ib)

The `iX` commands are similar in syntax with the `dX` commands except they access memory using I/O cycles instead of memory cycles. They follow the format:

```
<command> <address> [[1] bytelength]
```

By default, the bytelength is dependant on the command. Using `il` will return 32 bits of data, `iw` will return 16 bits, and `ib` will return 1 byte. All these lengths can be overridden, however. By re-typing the command with no arguments, PLXMon 99 will continue to display the memory locations using the size of the data retrieved as the increment size.



3.6.3.3 Writing Memory Via Memory Cycles, (el, ew, eb)

Again the syntax of the write using memory cycles is similar to the dX commands. Using e1 will write values as 32 bit wide objects, ew will write on 16 bit values, and eb will write a byte at a time.

They follow the format:

<command> <address> [INC increment] [value]

While the input address is required, both the value and the increment parameter are optional. By not entering the [value] parameter, the program will query you for data in interactive mode. Interactive mode allows the user to press the space bar between after typing the value he/she want to enter and PLXMon 99 will auto-increment the write address the size of the data being entered. Press the enter key to exit the command. By changing the INC parameter in the command line above, the auto-increment value can be changed. What follows is a brief example of how to use this command.

If the user types the command eb s0 INC 4, they wish to interactively write bytes to the location s0 (which is a system label, more on this in section 3.7.3.6). Every time the user enters a value and hits space the program will query for the previous address plus INC which is 4. On display of the memory range the user should see the following screen.

```

&eb s0 INC 4
80018000 01:a 02:b 03:c 04:d 05:e 06:f

&d1 s0 1 20
80018000: 0002000A 0000000B 0000000C 0000000D .....
80018010: 0000000E 0000000F 00000007 00000008 .....

```

Ready | Active Pane: Lower Pane | Communication Mode: PCI | PLXMon99.PLX

The 00: bytes that appear before each user entry are the previous values that are about to be overwritten by the user's data.

Figure 3-8. Interactive Mode

3.6.3.4 Writing Memory Via I/O Cycles, (ol, ow, ob)

The oX commands are simpler syntactically than the memory cycle writes. Each command writes a different sized data object to a port address. They require only two parameters:
<command> <address> <value>

3.6.3.5 The Pci Command

This command allows read/write access the PCI configuration registers. Its syntax is as follows:

pci <pci offset> [value]

To write to the required offset, just add the value to write; otherwise the value will be displayed as a 32-bit register value.

Note: the offset will be different depending on the Access mode: PCI or Serial.

3.6.3.6 The Quit Command

The quit command terminates the application PLXMon 99.

3.6.3.7 The Reg Command

The `reg` command allows users access to the PLX RDK's local register sets. Data can be read or written in 32 bit sizes at a given byte boundary. The syntax of the command is as follows:
`reg <register offset> [value]`

If a `[value]` is given, the command will write the data to the specified address.

3.6.3.8 The Repeat Command (r)

The repeat command can be used to make PLXMon 99 repeat the command types before the `r` a set number of times. Its syntax is as follows:
`[command] r [iterations]`

If the number of `iterations` is not given, then PLXMon 99 will execute the command indefinitely until the user hits a key. The command to be repeated must be in the same expression as the `r` command.

3.6.3.9 User Variables (vars)

PLXMon 99 creates some user labels as a mnemonic aid for common memory locations. These strings can be used interchangeably with the values they represent.

This table lists the variables set by PLXMon 99 and what memory ranges they represent.

Variable Name	Description	PCI or Serial
hbuf	User mapped region of the PCI common buffer.	PCI
PlxRegBase	Memory-mapped PLX RDK registers	PCI
PlxIoBase	I/O mapped PLX RDK registers	PCI
PlxLocalBusBase	Memory address which represents the Local Bus	PCI
s0	Local Space 0	PCI
s1	Local Space 1	PCI
s2	Local Space 2	PCI
s3	Local Space 3	PCI
pciVar	Local address base of PCI registers	Serial
lcrVar	Local address base of local configuration registers	Serial
rtrVar	Local address base of runtime registers	Serial
dmaVar	Local address base of DMA registers	Serial
mqrVar	Local address base of messaging queue registers	Serial

Figure 3-9. User Variables and their definitions

3.6.3.10 The Ver Command

Displays the version data contained in the PCI SDK software release. This version of PLXMon 99 is compatible only with PCI SDK v3.0. This command is usable only during PCI mode.

3.7 Print, Print Preview, and Print Setup

The print commands (`Print`, `Print Preview`, `Print Setup...`) can be found under the `File` pull-down menu. These selections enable the user to create a formatted picture of the display window. Print preview will allow you to see the formatted screen before you print.

3.8 Register Access/Register Sets

The contents of registers can be represented in one of two ways in PLXMon 99. Usually when a full 32-bit register is being displayed, it is shown in an edit box in hexadecimal format (*the 0x prefix is implied*). These boxes, if not grayed-out, can be modified by typing in new values. The value(s) will be updated when the user closes the window or when the cursor is moved to another edit box.

Check boxes are also used to display and change individual bits on a register. If the check box is on a main dialog window then a state change is immediate. If the check box is in a Details dialog with other checkboxes, then the changes are made only upon closing the dialog box.

Note: Please refer to Appendices A, B, and C for specific details on the IOP 480, PCI 9054 and PCI 9080 device registers, respectively.

3.9 The Reset Button

Found on the taskbar, the  reset button signals the PLX RDK that is currently selected to reset itself. This action can be taken during both PCI and Serial modes.

Note: The method used to reset the board is customized to work with the PLX RDKs. This feature should not be used on devices/boards that do not support the reset algorithm. Consult the PCI SDK BSP source code for complete information on the specific algorithm(s) used with various PLX RDK boards.

3.10 Selecting Devices

The Select A Device menu item, shown in Figure 3-11, is used to select a PCI device that you want to access. The pointer points to the currently selected device.

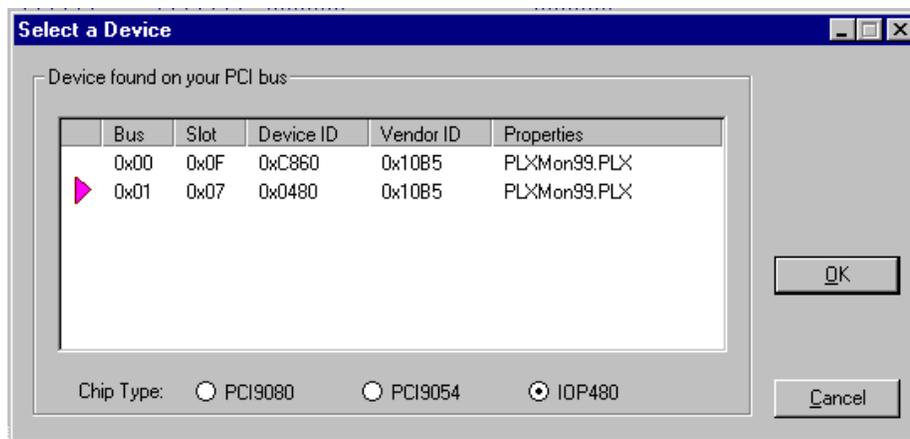


Figure 3-10. Device Select Dialog Box

Figure 3-10 lists two boards, the CompactPCI 9054RDK-860 (Device ID=0xC860) and the IOP 480RDK (Device ID=0x0480). To select a new device, move the highlight to the desired PCI device by using either the mouse or the cursor arrow keys, and click the OK button or press the Enter key. The pointer will not move to the new selection until OK button is pressed or until the item is double-clicked. The chip type radio button indicates which PLX device is present on the selected device.

3.11 Serial Configuration

PLXMon 99 offers the capability to communicate with the PLX device through the serial port. To do so you must configure the appropriate serial port settings as shown in the figure below.

Communication ports COM1 through COM4 are supported. Baud rates 9600, 19200, 38400, and 57600 are supported.

Note: All PLX RDKs should be configured to 38400 baud, 8 Data Bits, No Parity, 1 Stop Bit, and no Flow Control.

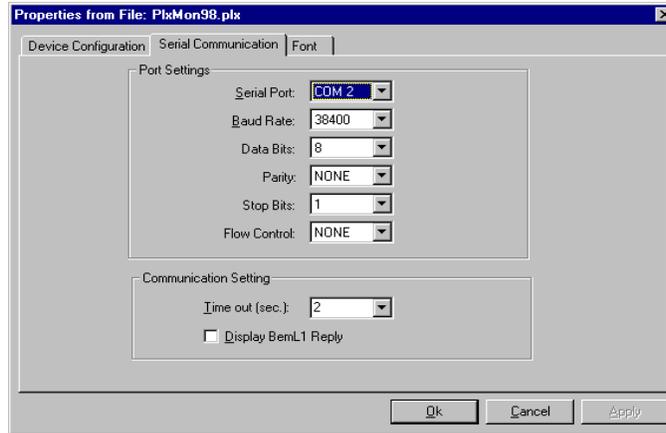


Figure 3-11. Serial Communications Properties Dialog Box

3.12 Serial EEPROM Access

Pressing the Serial EEPROM  button on the toolbar of PLXMon 99 will start the EEPROM dialog box. The EEPROM screen for IOP 480RDK is shown in Figure 3-12:

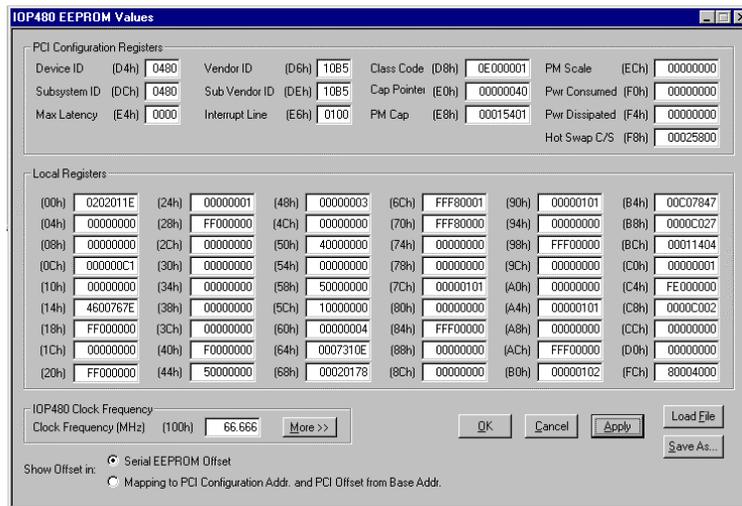


Figure 3-12. Serial EEPROM Access Screen

PLXMon 99 supports saving the EEPROM values to a file and also load values from a file.

Note: Having the wrong Serial EEPROM type selected in the Device Config. Menu can cause PLXMon 99 to read/write invalid data.



This page intentionally left blank.

4. The IOP 480 Register Set

Each of the IOP 480 register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers if necessary.

4.1 The Register Group Dialog Boxes

The PLXMon 99's toolbar for IOP 480 contains seven buttons for register accesses. They are PCI Configuration Registers (PCR), Local Configuration Registers (LCR), RunTime Registers (RTR), DMA Registers (DMA), Messaging Queue Registers (MQR), Memory Controller Registers (MCR), and IOP 480CPU registers (IOP480 CPU).

4.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in Figure 4-1, on the PCI Configuration Registers dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and check boxes, indicate the current settings of the register bit fields. To update the contents of the dialog box push the Refresh button.

Vendor ID	(00h)	1085	Device ID	(02h)	0480	Command	(04h)	0117	Status	(06h)	0290
Revision ID	(08h)	01	Class Code	(09h)	0E0000	Cache Line Size	(0Ch)	08	Latency	(0Dh)	20
Header Type	(0Eh)	00	Build-In ST	(0Fh)	00	<input type="checkbox"/> BIST					
Base Address 0	(10h)	FB000000									
Base Address 1	(14h)	FA000000	Details...								
Base Address 2	(18h)	00000000	Details...								
Base Address 3	(1Ch)	00000000									
Base Address 4	(20h)	00000000									
Base Address 5	(24h)	00000000									
CardBus CIS Ptr	(28h)	00000000	Sub Vendor ID	(2Ch)	1085	SubSystem ID	(2Eh)	0480			
Expansion RDM	(30h)	00000000	<input type="checkbox"/> Address Decode Enable			New Capability Pointer	(34h)	00000040			
Interrupt Line	(3Ch)	08	Interrupt Pin	(3Dh)	01	Minimum Grant	(3Eh)	00	Max Latency	(3Fh)	00
Power Mgmt Capability ID	(40h)	01	Power Mgmt Next Item Ptr	(41h)	54	Power Mgmt Capabilities	(42h)	0001	Details...		
Power Mgmt Control Status	(44h)	00000000	Details...								
Power Mgmt DataScale Value	(48h)	00000000	Details...								
Power Consumed	(4Ch)	00000000	Details...								
Power Dissipated	(50h)	00000000	Details...								
Hot Swap ID	(54h)	00	Hot Swap Next Cap Pointer	(55h)	58	HS Control/Status	(56h)	00	Details...		
VPD ID	(58h)	03	Next Cap Pointer	(59h)	00	VPD Address	(5Ah)	0100			
VPD Data	(5Ch)	00055000									

Figure 4-1. PCI Configuration Registers Dialog Box

4.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes, in Figure 4-2, respectively. The size text box reflects the value (in bytes) of the associated register. The memory size is calculated from the corresponding register value and cannot be modified directly. To change the memory size, modify the associated register.

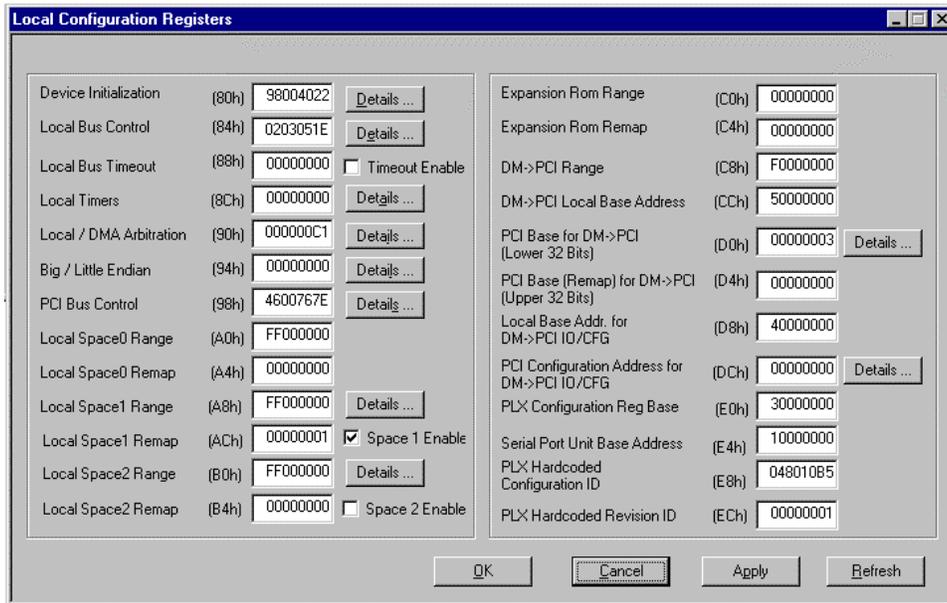


Figure 4-2. Local Configuration Registers Dialog Box

4.1.3 The Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays the current register values. Changing the contents of any register can modify all register values. See figure below:

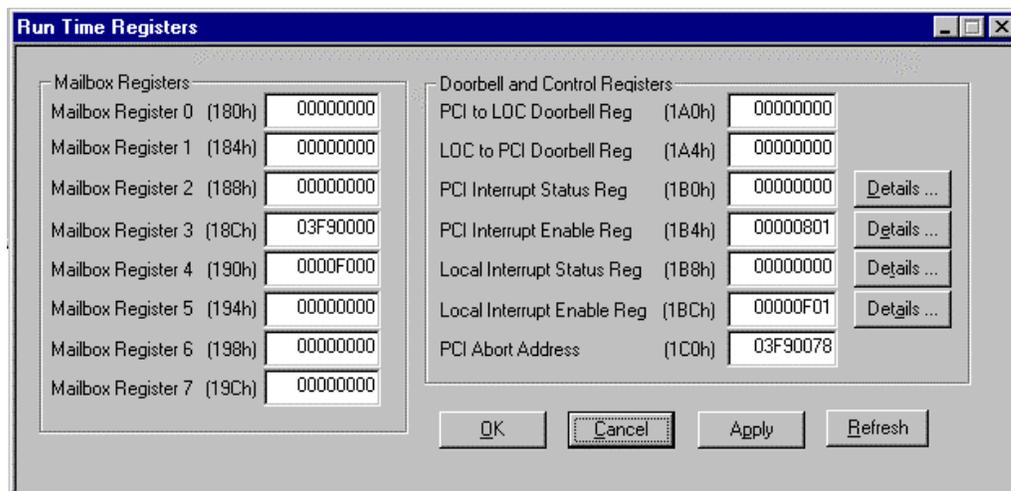


Figure 4-3. Runtime Register Group Dialog Box

The Interrupt Enable/Status Register Dialog Box

There are four Details buttons for Interrupt Enable/Status Registers. These are for PCI and Local Interrupt Status and Enable Registers. The Interrupt Enable/Status Register Dialog Boxes provides information on the current value of the Interrupt Enable/Status register. The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

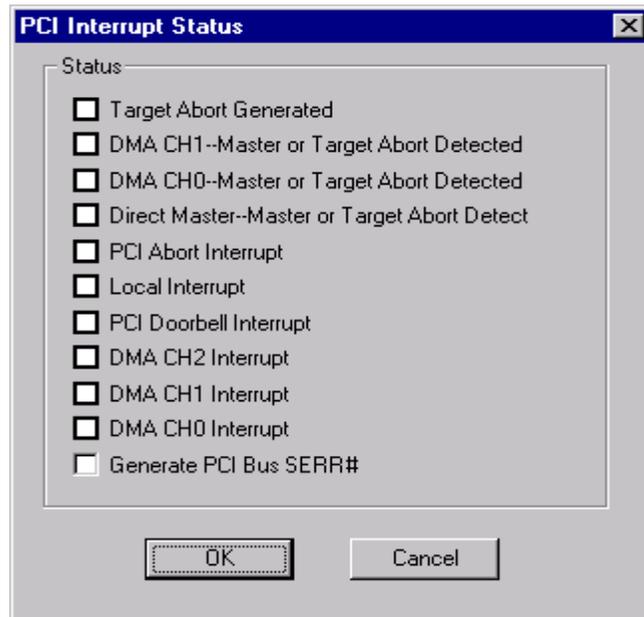


Figure 4-4. Interrupt Enable /Status Register Dialog Box

4.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for all three DMA channels. See figure below.

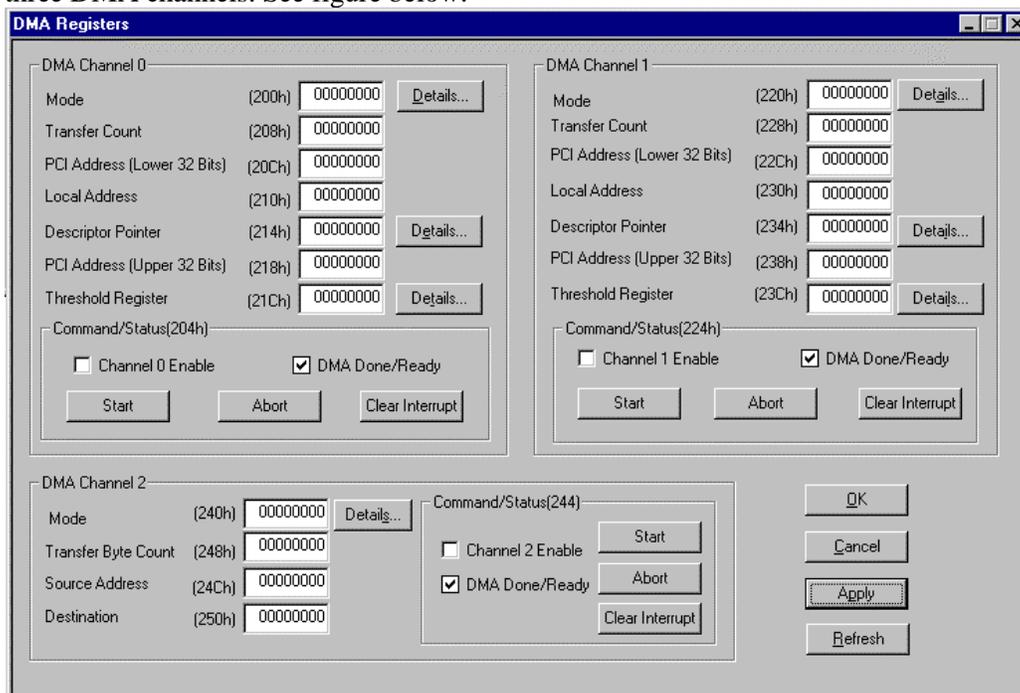


Figure 4-5. DMA Register Group Dialog Box

The Start and Abort Transfer buttons initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Clear Interrupt button resets the interrupts to their default state. The Channel Enable bit enables DMA transfers and activates the Start, Abort, and Clear Interrupt buttons. The DMA Done/Ready bit indicates the DMA engine status.

4.1.5 The Messaging Queue Register Group Dialog Box

The Messaging Queue Register Group dialog box contains the current values for the Messaging FIFO Registers as shown below:

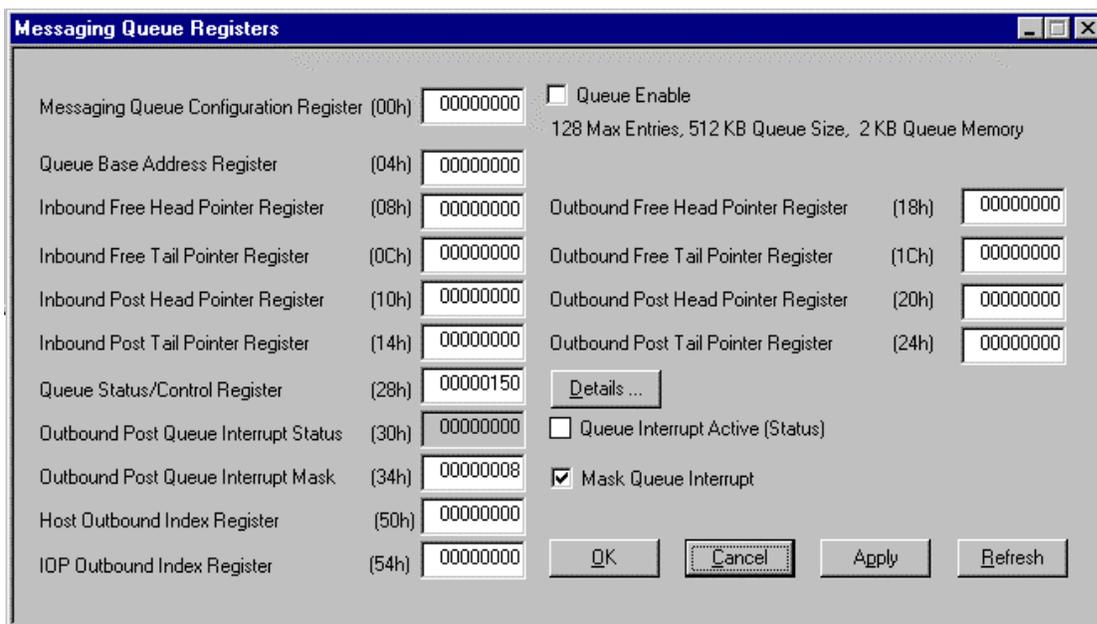


Figure 4-6. Messaging Queue Register Group

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified.

4.1.6 IOP 480 Memory Controller register Group Dialog Box

The Memory Controller Register Group dialog box contains the current values for the IOP 480's integrated memory controller. The memory controller register group is divided into five subgroups (see figure below), namely, LCS0, LCS1, LCS2, LCS3, and DRAM Control registers. For more information, please consult the IOP 480 data book.

Note: (LCS = Local Chip Select)

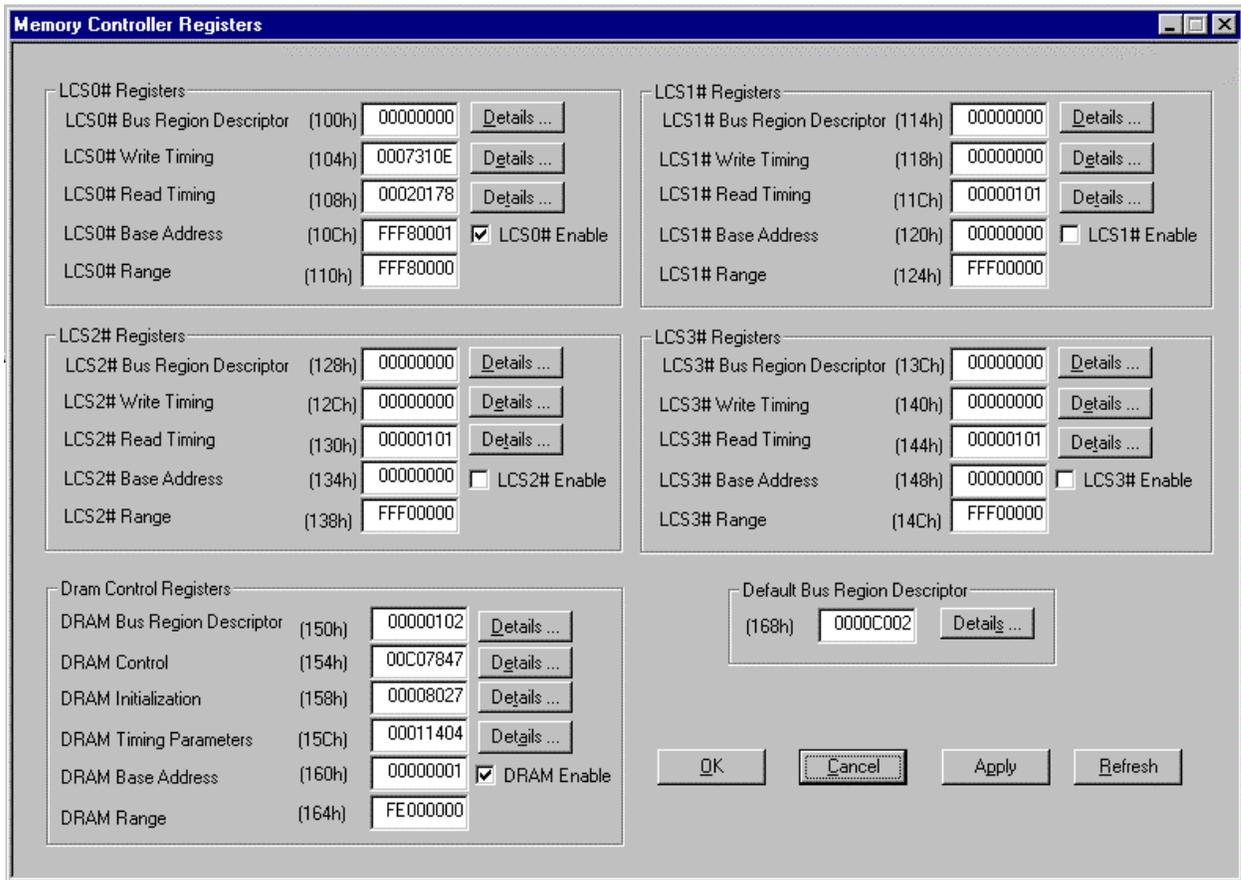


Figure 4-7. IOP 480 Memory Controller Registers

4.1.7 IOP 480 CPU Registers Group Dialog Box

The IOP 480 CPU Register Group dialog box contains the current values for the IOP 480 CPU registers. These are divided into two groups, General Purpose Registers and Special Purpose Registers. Please refer to the IOP 480 data book for more information on these registers. See figure below:

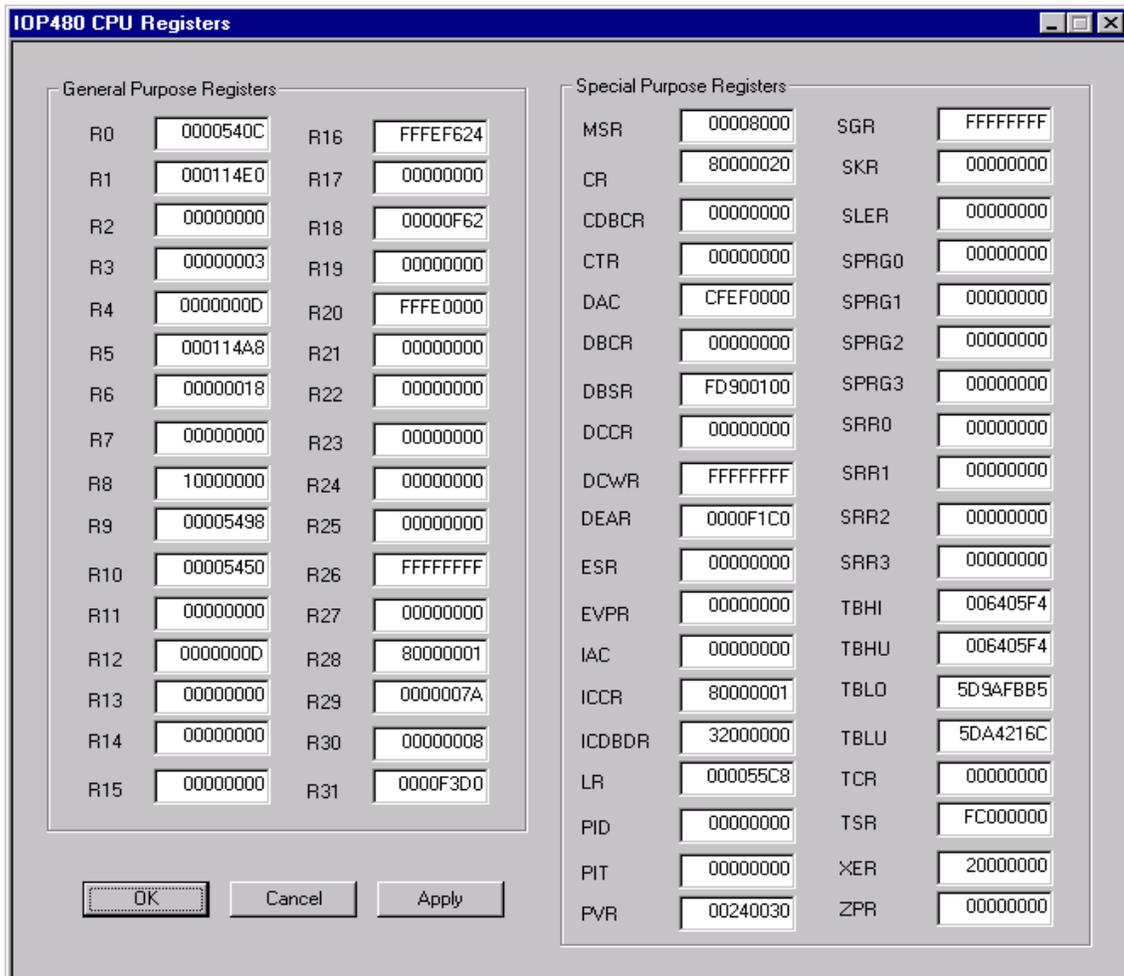


Figure 4-8. IOP 480 CPU Registers Group

4.1.8 IOP 480 EEPROM Values

The following figure shows the default values of the EEPROM on the IOP 480 RDK board. These values may be used as the default values. This dialog box allows one to save the EEPROM values to a file and load values from a file. Clicking on the <OK> and <Apply> button will write the current values to your EEPROM.

Note: You should be very careful before writing any value to the EEPROM. Wrong EEPROM values can cause your board to crash and it may not reboot. You should save the default values to a file on a floppy diskette as a backup.

IOP480 EEPROM Values

PCI Configuration Registers

Device ID (D4h)	0480	Vendor ID (D6h)	10B5	Class Code (D8h)	0E000001	PM Scale (ECh)	00000000
Subsystem ID (DCh)	0480	Sub Vendor ID (DEh)	10B5	Cap Pointer (E0h)	00000040	Pwr Consumed (F0h)	00000000
Max Latency (E4h)	0000	Interrupt Line (E6h)	0100	PM Cap (E8h)	00015401	Pwr Dissipated (F4h)	00000000
						Hot Swap C/S (F8h)	00025800

Local Registers

(00h)	0202011E	(24h)	00000001	(48h)	00000003	(6Ch)	FFF80001	(90h)	00000101	(B4h)	00C07847
(04h)	00000000	(28h)	FF000000	(4Ch)	00000000	(70h)	FFF80000	(94h)	00000000	(B8h)	0000C027
(08h)	00000000	(2Ch)	00000000	(50h)	40000000	(74h)	00000000	(98h)	FFF00000	(BCh)	00011404
(0Ch)	000000C1	(30h)	00000000	(54h)	00000000	(78h)	00000000	(9Ch)	00000000	(C0h)	00000001
(10h)	00000000	(34h)	00000000	(58h)	50000000	(7Ch)	00000101	(A0h)	00000000	(C4h)	FE000000
(14h)	4600767E	(38h)	00000000	(5Ch)	10000000	(80h)	00000000	(A4h)	00000101	(C8h)	0000C002
(18h)	FF000000	(3Ch)	00000000	(60h)	00000004	(84h)	FFF00000	(A8h)	00000000	(CCh)	00000000
(1Ch)	00000000	(40h)	F0000000	(64h)	0007310E	(88h)	00000000	(ACh)	FFF00000	(D0h)	00000000
(20h)	FF000000	(44h)	50000000	(68h)	00020178	(8Ch)	00000000	(B0h)	00000102	(FCh)	80004000

IOP480 Clock Frequency

Clock Frequency (MHz) (100h) 66.666 More >>

OK Cancel Apply Load File Save As...

Show Offset in:

- Serial EEPROM Offset
- Mapping to PCI Configuration Addr. and PCI Offset from Base Addr.

Figure 4-9. IOP 480 EEPROM values



By default, the IOP 480RDK board is configured for 66.66MHz frequency. If your board is designed to work at a different frequency, you can change the frequency in PLXMon 99 EEPROM screen and you will get a pop up message window as shown below:

The screenshot shows the 'IOP480 EEPROM Values' dialog box. It is divided into several sections:

- PCI Configuration Registers:** Contains fields for Device ID (0480), Vendor ID (10B5), Class Code (0E000001), PM Scale (00000000), Subsystem ID (0480), Sub Vendor ID (10B5), Cap Pointer (00000040), Pwr Consumed (00000000), Max Latency (0000), Interrupt Line (0100), PM Cap (00015401), Pwr Dissipated (00000000), and Hot Swap C/S (00025800).
- Local Registers:** A table of 16-bit registers from 00h to FFh. A pop-up window titled 'PLXMon 99' is overlaid on this section. The pop-up contains a yellow warning icon and the following text:

NOTE:
 If using SDRAM and assuming a refresh rate of 15.625 us, the interval between Refresh cycles is calculated as follows:
 $\text{Refresh Interval}(515) = \text{Refresh rate}(15.625 \text{ us}) * \text{Clock Frequency}(33.000 \text{ MHz})$
 The value of 0x00A03847 for EEPROM at EEPROM offset 0xB4 is recommended.
 If you are using different SDRAM, please verify the Refresh rate of your SDRAM.

 An 'OK' button is at the bottom of the pop-up.
- IOP480 Clock Frequency:** A field for 'Clock Frequency (MHz)' is set to 33.000, with a 'More >>' button next to it.
- Show Offset in:** Two radio buttons: 'Serial EEPROM Offset' (selected) and 'Mapping to PCI Configuration Addr. and PCI Offset from Base Addr.'.
- Buttons:** 'OK', 'Cancel', 'Apply', 'Load File', and 'Save As...'.

Note: It is recommended that you provide the correct value for the type of SDRAM you are using on your board.

5. The PCI 9054 Register Set

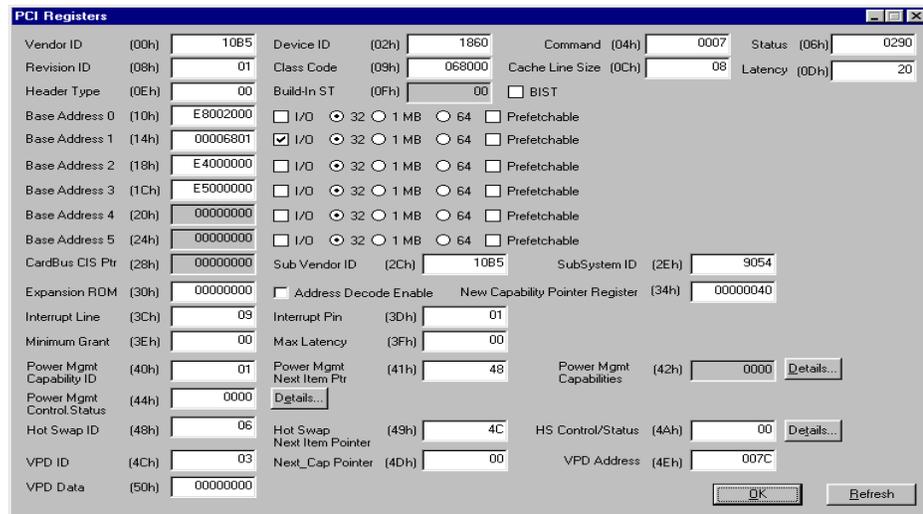
Each of the PCI 9054's register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers when required.

5.1 The Register Group Dialog Boxes

The PLXMon 99's toolbar contains five buttons for register accesses. They are for PCI Configuration Registers (PCR), Local Configuration Registers (LCR), RunTime Registers (RTR), DMA Registers (DMA), and Messaging Queue Registers (MQR).

5.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in the PCI Configuration Registers dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and check boxes, also in Figure 5-1, indicate the current settings of the register bit fields. To update the contents of the dialog box push the Refresh button.



Register Name	Address	Value	Register Name	Address	Value	Register Name	Address	Value		
Vendor ID	(00h)	10B5	Device ID	(02h)	1860	Command	(04h)	0007		
Revision ID	(08h)	01	Class Code	(09h)	068000	Cache Line Size	(0Ch)	08		
Header Type	(0Eh)	00	Build-In ST	(0Fh)	00	Status	(06h)	0290		
Base Address 0	(10h)	E8002000	<input type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable	Latency	(0Dh)	20
Base Address 1	(14h)	00006801	<input checked="" type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable			
Base Address 2	(18h)	E4000000	<input type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable			
Base Address 3	(1Ch)	E5000000	<input type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable			
Base Address 4	(20h)	00000000	<input type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable			
Base Address 5	(24h)	00000000	<input type="checkbox"/> I/O	<input checked="" type="radio"/> 32	<input type="radio"/> 1 MB	<input type="radio"/> 64	<input type="checkbox"/> Prefetchable			
CardBus CIS Ptr	(28h)	00000000	Sub Vendor ID	(2Ch)	10B5	SubSystem ID	(2Eh)	9054		
Expansion ROM	(30h)	00000000	<input type="checkbox"/> Address Decode Enable	New Capability Pointer Register	(34h)	00000040				
Interrupt Line	(3Ch)	09	Interrupt Pin	(3Dh)	01					
Minimum Grant	(3Eh)	00	Max Latency	(3Fh)	00					
Power Mgmt Capability ID	(40h)	01	Power Mgmt Next Item Ptr	(41h)	48	Power Mgmt Capabilities	(42h)	0000		
Power Mgmt Control Status	(44h)	0000					Details...			
Hot Swap ID	(48h)	06	Hot Swap Next Item Pointer	(49h)	4C	HS Control/Status	(4Ah)	00		
VPD ID	(4Ch)	03	Next_Cap Pointer	(4Dh)	00			Details...		
VPD Data	(50h)	00000000			VPD Address	(4Eh)	007C			

Figure 5-1. PCI Configuration Registers Dialog Box

Power Management Capabilities

This dialog box seen in Figure 5-2 displays Power Management setup attributes that are read only, or that can only be modified from the IOP side. If this dialog box is opened in Serial Mode, then certain values can be changed.

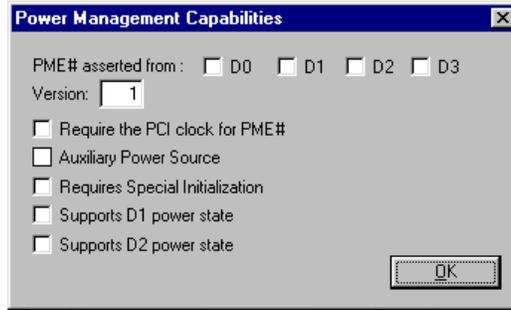


Figure 5-2. Power Management Capabilities Dialog Box

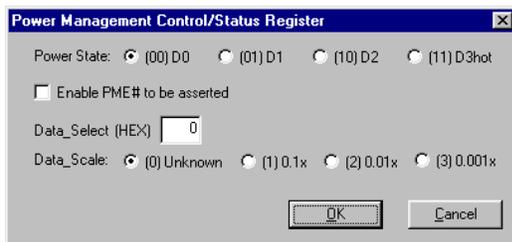


Figure 5-3. Power Management CSR Dialog Box

Power Management Control/Status Register

In Figure 5-3 are the bits that handle the operation of Power Management on the PCI 9054.

Hot Swap Control/Status Register

The Control and Status bits for Hot Swapping are found here in Figure 5-4. All of these values can only be written from the PCI side.

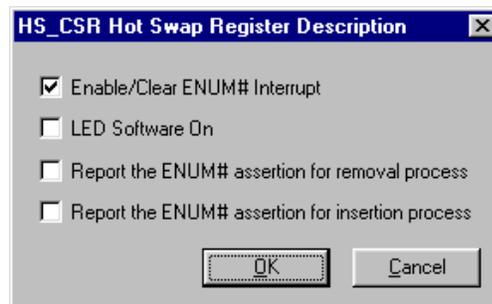


Figure 5-4. Hot Swap CSR Dialog Box

5.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes as seen in Figure 5-5. The size text box reflects the value (in bytes) of the associated register. The memory size is calculated from corresponding register values and cannot be modified directly. To change the memory size, modify the associated register.

Seven registers within the Local Configuration Register Group have a more detailed dialog box and are as follows:

- The Mode/DMA Arbitration dialog box
- The Endian Descriptor dialog box
- The Miscellaneous Control Register dialog box
- The Region 0/Exp ROM dialog box
- The DM PCI Remap dialog box
- The DM Config IO Address dialog box
- The Region 1 Descriptor dialog box

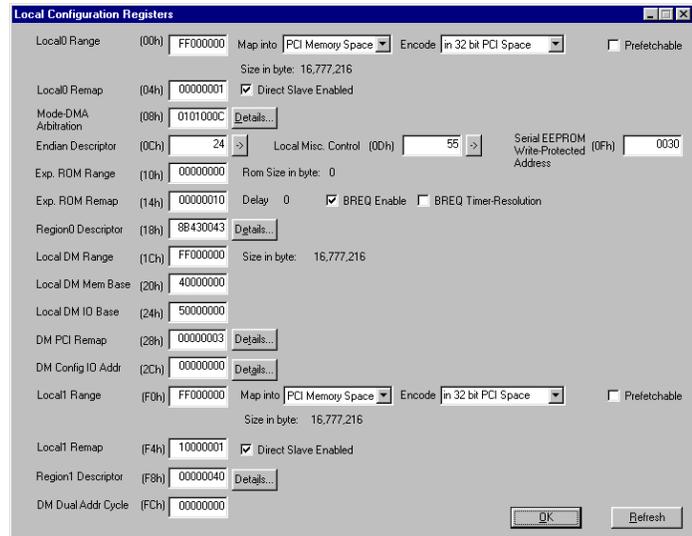


Figure 5-5. Local Configuration Registers Dialog Box

The Mode/Arbitration Dialog Box

The Mode/Arbitration dialog box provides information on the current value Local/DMA Arbitration register and allows modification of that value (see Figure 5-6).

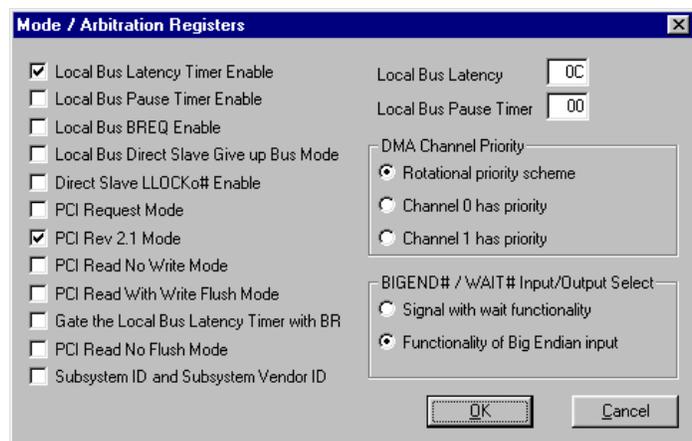
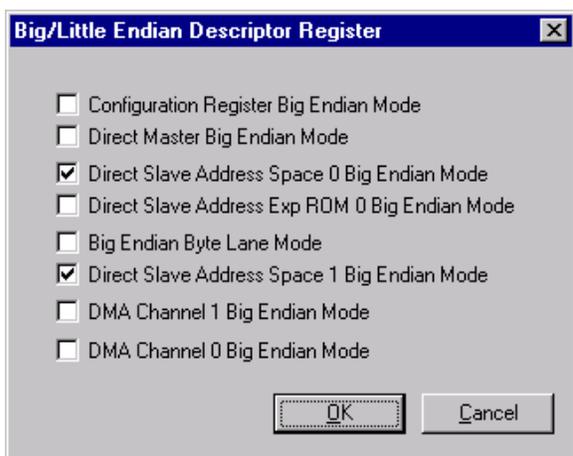


Figure 5-6. Mode/Arbitration Dialog Box



Endian Descriptor Dialog Box

The Endian Descriptor dialog box provides information on the current value of the Big/Little Endian Descriptor register and allows modification of that value (see Figure 5-7).

Figure 5-7. Endian Descriptor Dialog Box

Local Miscellaneous Control Register Dialog Box

This dialog box contains bit controls for the miscellaneous functions of the PCI 9054 (see Figure 5-8).

These functions include:

- Base Address Register 1 support
- Init Done bit signal to BIOS
- Direct Master Enables
- Error interrupt Masks

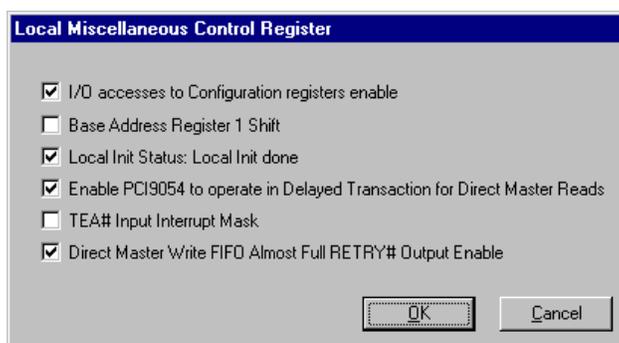


Figure 5-8. Local Miscellaneous Control Register Dialog Box

The Local Space 0/Exp ROM Dialog Box

The Region 0 descriptor provides information on the current value of the Local Address Space 0/Expansion ROM Bus Region Descriptor register and allows modification of that value

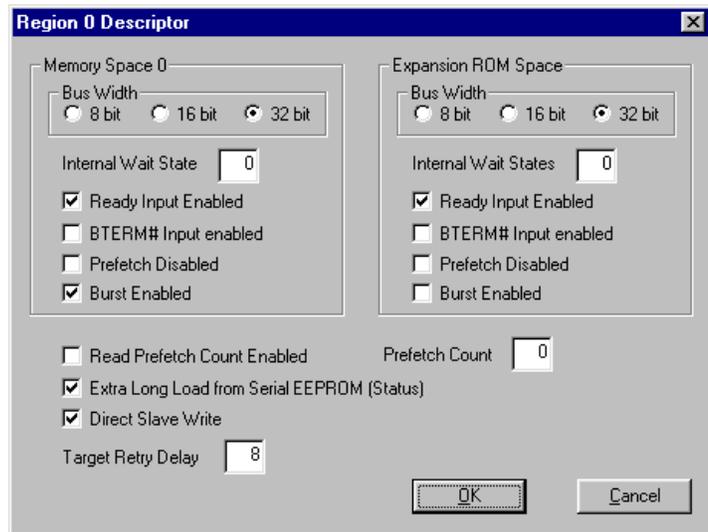


Figure 5-9. Region 0/ROM Descriptor Dialog Box

The DM PCI Remap Dialog Box

The DM PCI Remap dialog box provides information on the current value of the PCI Base Address (Remap) Register for Direct Master to PCI Memory and allows modification of that value (see Figure 5-10).

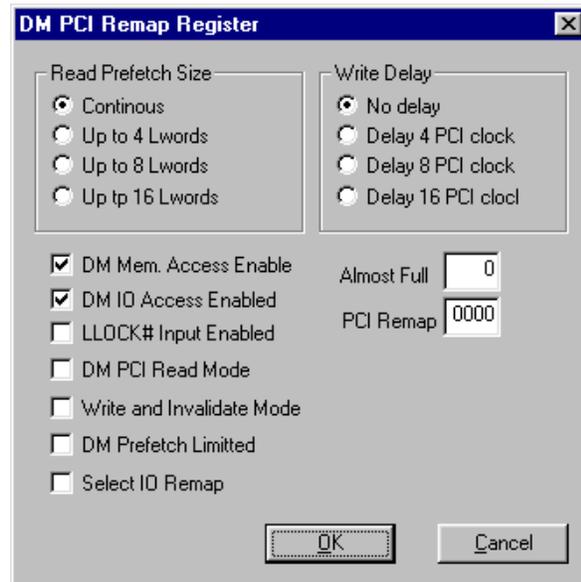
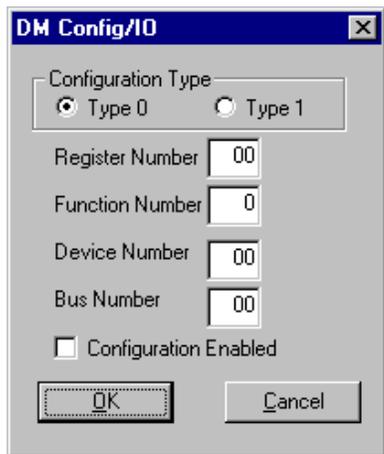


Figure 5-10. Direct Master Remap Dialog Box



The DM Configuration I/O Address Dialog Box

The DM Configuration I/O Address dialog box provides information on the current value of the PCI configuration Address Register for Direct Master to PCI I/O-CFG and allows modification of that value (see Figure 5-11).

Figure 5-11. Direct Master Config. Dialog box

The Local Space 1 Dialog Box

The Region 1 dialog box provides information on the Local Address Space 1 Bus Region Descriptor register and allows modification of that value (see Figure 5-12).

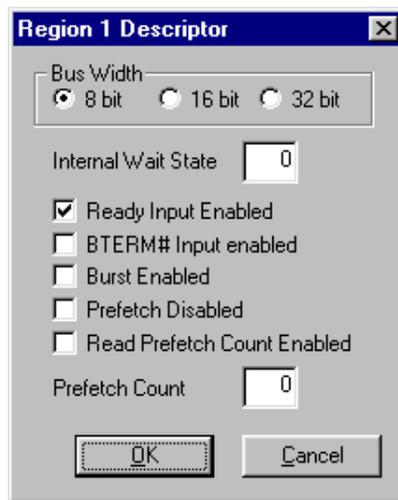


Figure 5-12. Region 1 Descriptor Dialog Box

5.1.3 The Runtime Register Group Dialog Box

The Run Time Register Group dialog box displays the current register values. All register values can be modified by changing the contents of any register (see Figure 5-13).

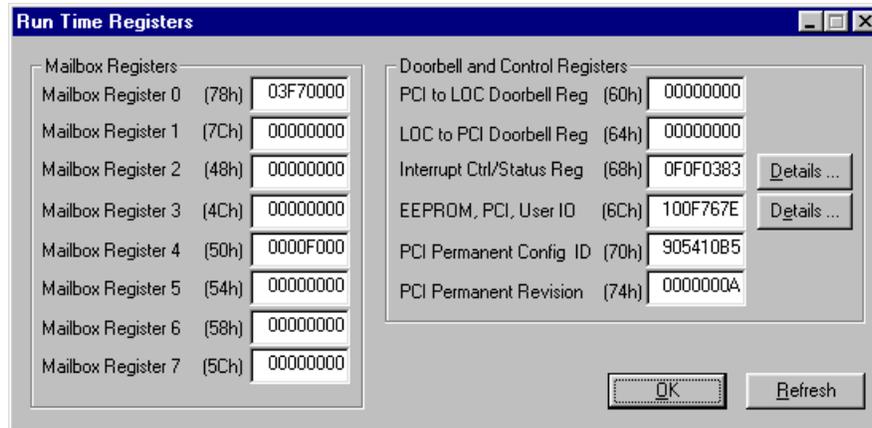


Figure 5-13. Run Time Registers Dialog Box

The Interrupt Control/Status Register Dialog Box

The Interrupt Control/Status Register Dialog Box provides information on the current value of the Interrupt Control/Status register.

The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

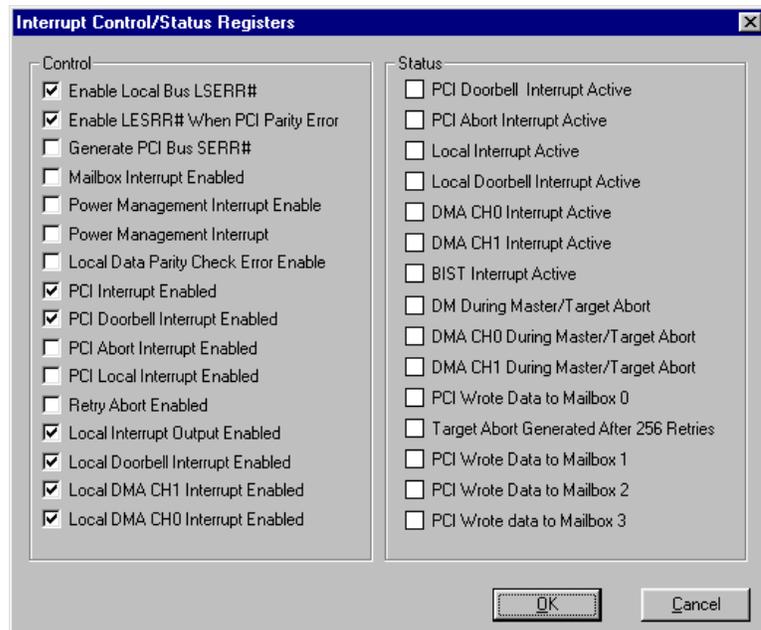


Figure 5-14. Interrupt Control and Status Dialog Box

The EEPROM, PCI, User IO Dialog Box

The EEPROM, PCI, User IO dialog box provides information on the current contents of the EEPROM Control, PCI Command Codes, User I/O Control, Init Control Register and allows modification of that value (see Figure 5-15). The Status section contained in this dialog box contains values that cannot be modified.

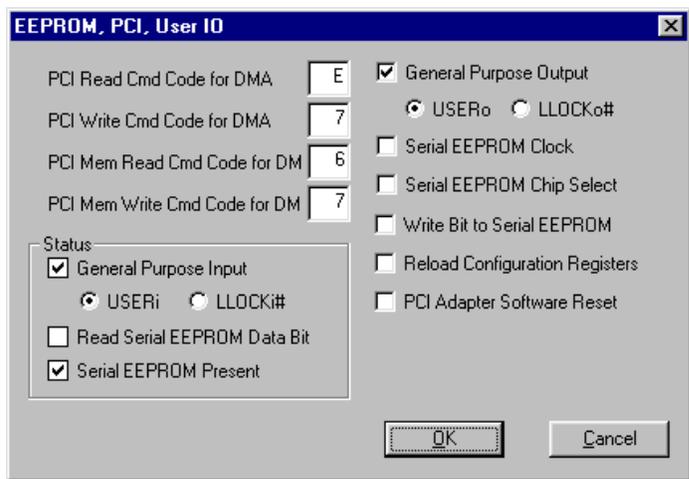


Figure 5-15. EEPROM PCI User IO Dialog Box

5.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for both DMA channels (see Figure 5-16).

The Start and Abort Transfer buttons initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Channel Enable bit enables DMA transfers and activates the Start, Abort, and Clear Interrupt buttons.

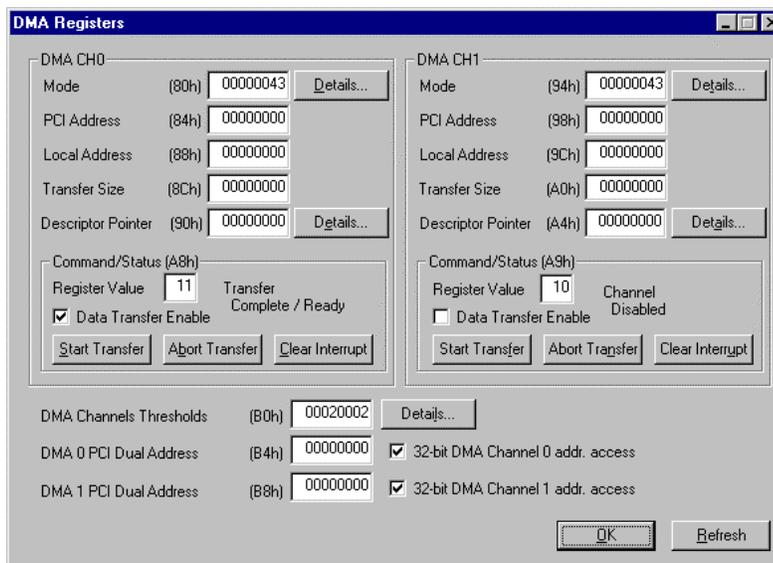
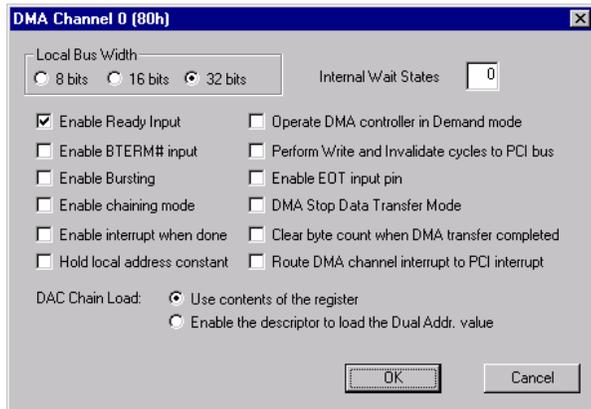


Figure 5-16. Local DMA Registers Dialog Box

The DMA Mode Dialog Box



The DMA Mode dialog box provides information on the current value the DMA Channel's Mode Register and allows modification of that value (see Figure 5-17).

The Descriptor Pointer Dialog Box

The Descriptor Pointer dialog box provides information on the current value of the DMA Channel's Descriptor Pointer Register and allows modification of that value (see Figure 5-18).

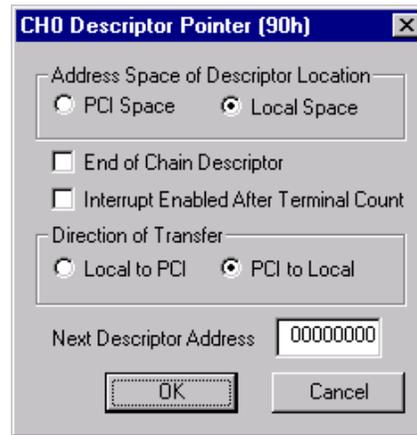


Figure 5-18. DMA Descriptor Pointer Dialog Box

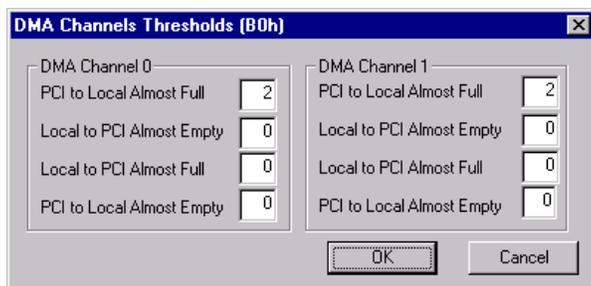


Figure 5-19. DMA Thresholds Dialog Box

The DMA Channels Threshold Dialog Box

The DMA Channels Threshold dialog box provides information on the current value of the DMA Threshold Register and allows modification of that value (see Figure 5-19).

5.1.5 The Messaging FIFO Register Group Dialog Box

The Messaging FIFO Register Group dialog box contains the current values for the Messaging FIFO Registers as shown in Figure 5-20.

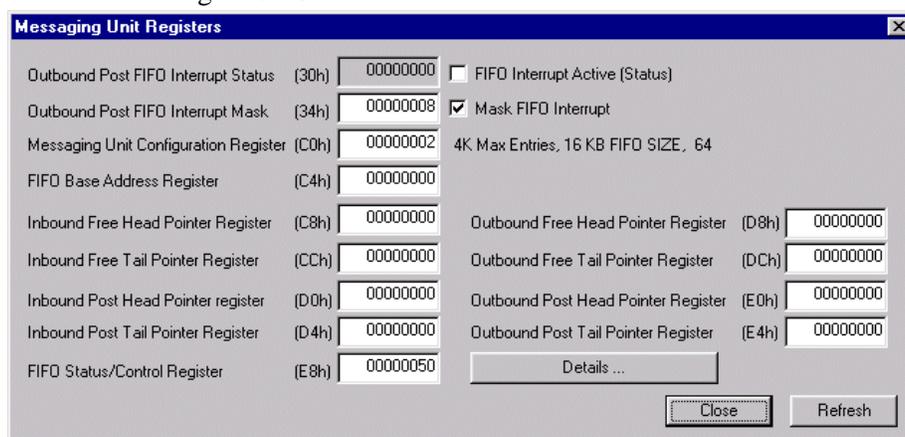


Figure 5-20. Messaging Unit Registers Dialog Box

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified.

The FIFO Status/Control Register Dialog Box

The FIFO Status/Control Register dialog box provides information on the current value of the Queue Status/Control Register and allows modification of that value (see Figure 5-21).

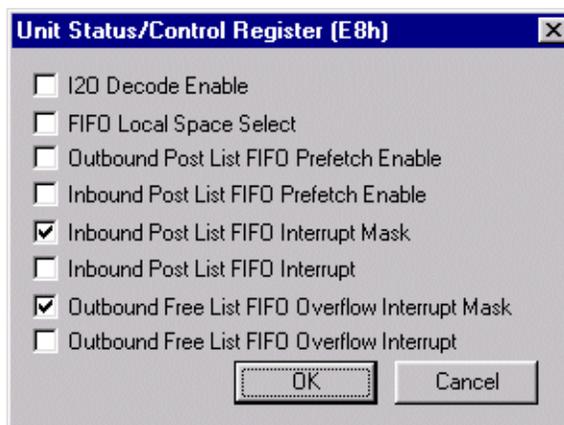


Figure 5-21. Status/Control Register Dialog Box

6. The PCI 9080 Register Set

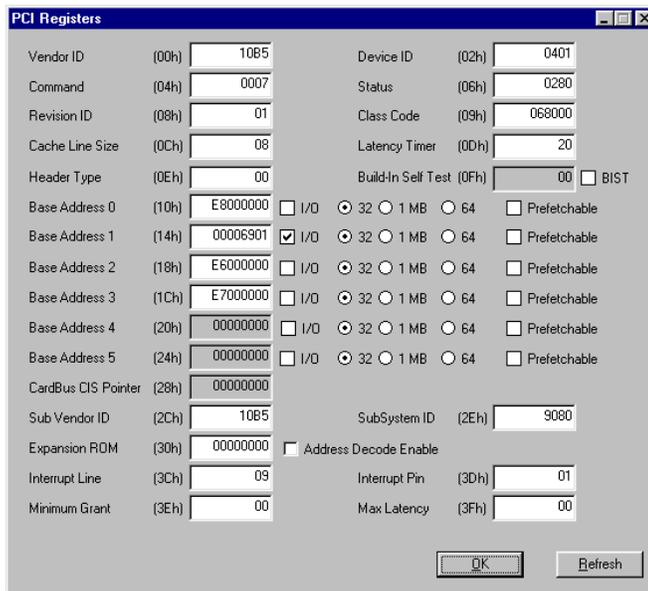
Each of the PCI 9080's register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers if necessary.

6.1 The Register Group Dialog Boxes

The PLXMon 99's toolbar contains five buttons for register accesses. They are for PCI Configuration Registers (PCR), Local Configuration Registers (LCR), RunTime Registers (RTR), DMA Registers (DMA), and Messaging Queue Registers (MQR).

6.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in Figure 6-1, on the PCI Configuration Registers dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and check boxes, indicate the current settings of the register bit fields. To update the contents of the dialog box push the Refresh button.



Register Name	Hex Address	Value	Options
Vendor ID	(00h)	10B5	
Device ID	(02h)	0401	
Command	(04h)	0007	
Status	(06h)	0280	
Revision ID	(08h)	01	
Class Code	(09h)	068000	
Cache Line Size	(0Ch)	08	
Latency Timer	(0Dh)	20	
Header Type	(0Eh)	00	
Build-In Self Test	(0Fh)	00	<input type="checkbox"/> BIST
Base Address 0	(10h)	E8000000	<input type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
Base Address 1	(14h)	00006901	<input checked="" type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
Base Address 2	(18h)	E6000000	<input type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
Base Address 3	(1Ch)	E7000000	<input type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
Base Address 4	(20h)	00000000	<input type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
Base Address 5	(24h)	00000000	<input type="checkbox"/> I/O <input checked="" type="radio"/> 32 <input type="radio"/> 1 MB <input type="radio"/> 64 <input type="checkbox"/> Prefetchable
CardBus CIS Pointer	(28h)	00000000	
Sub Vendor ID	(2Ch)	10B5	
SubSystem ID	(2Eh)	9080	
Expansion ROM	(30h)	00000000	<input type="checkbox"/> Address Decode Enable
Interrupt Line	(3Ch)	09	
Interrupt Pin	(3Dh)	01	
Minimum Grant	(3Eh)	00	
Max Latency	(3Fh)	00	

Figure 6-1. PCI Configuration Registers Dialog Box

6.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes, in Figure 6-2, respectively. The size text box reflects the value (in bytes) of the associated register. The memory size is calculated from the corresponding register value and cannot be modified directly. To change the memory size, modify the associated register.

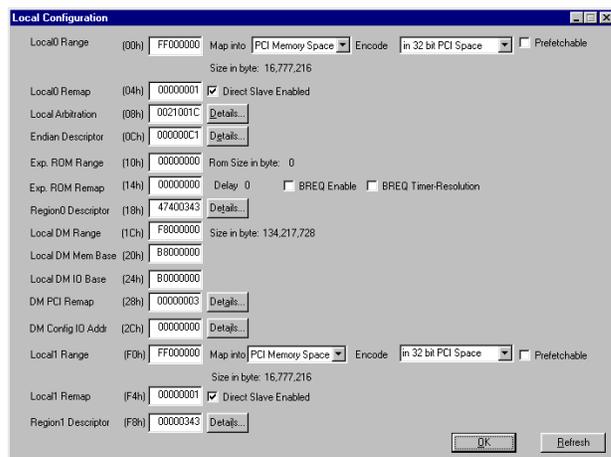


Figure 6-2. Local Configuration Registers Dialog Box

Six registers within the Local Configuration Register Group have a more detailed dialog box and are as follows:

- The Mode/Arbitration dialog box;
- The Endian Descriptor dialog box;
- The Space 0/Exp ROM dialog box;
- The DM PCI Remap dialog box;
- The DM Config IO Address dialog box; and,
- The Space 1 dialog box.

The Mode/Arbitration Dialog Box

The Mode/Arbitration dialog box provides information on the current value Local/DMA Arbitration registers and allows modification of that value (see Figure 6-3).

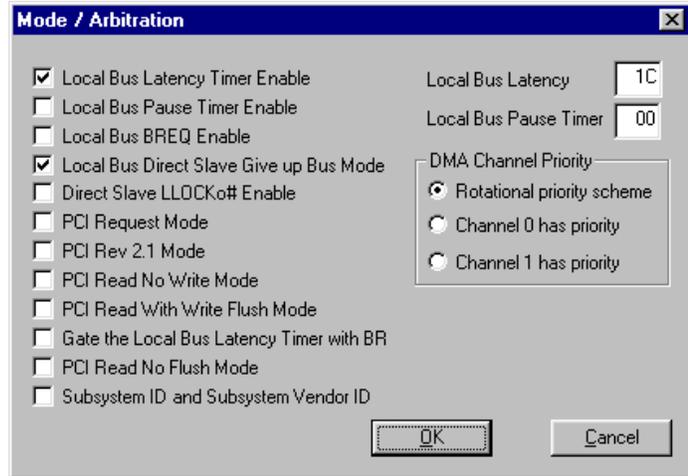


Figure 6-3. Direct Master Mode / Arbitration Dialog box

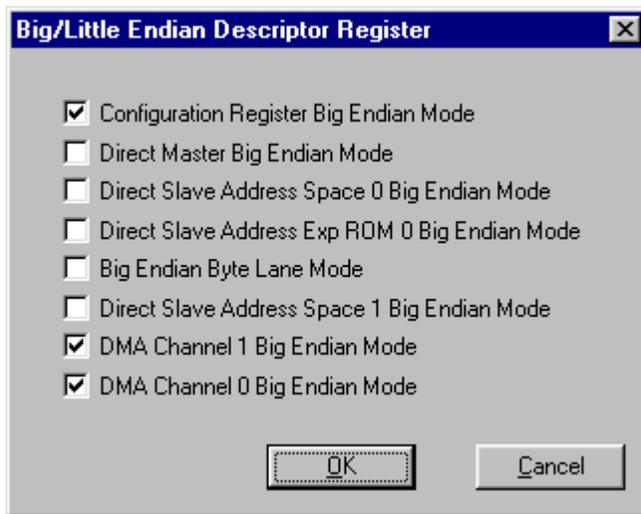


Figure 6-4. Endian Descriptor Dialog Box

Endian Descriptor Dialog Box

The Endian Descriptor dialog box provides information on the current value of the Big/Little Endian Descriptor register and allows modification of that value (see Figure 6-4).

The Local Space 0/Exp ROM Dialog Box

The Region 0 Descriptor dialog box provides information on the current value of the Local Address Space 0/Expansion ROM Bus Region Descriptor register and allows modification of that value (see Figure 6-5).

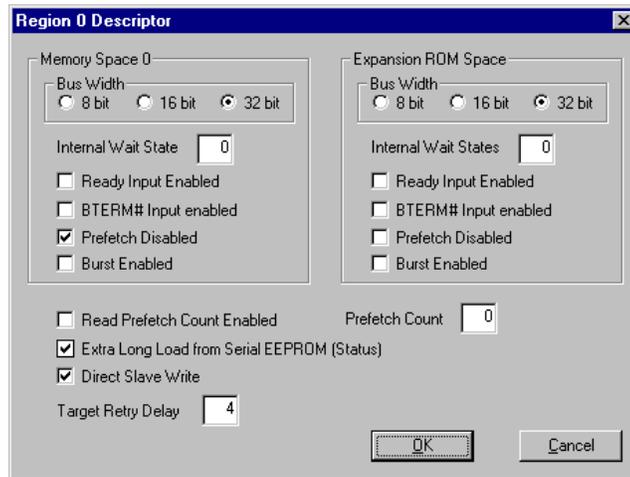


Figure 6-5. Local Space 0/Exp ROM Dialog Box

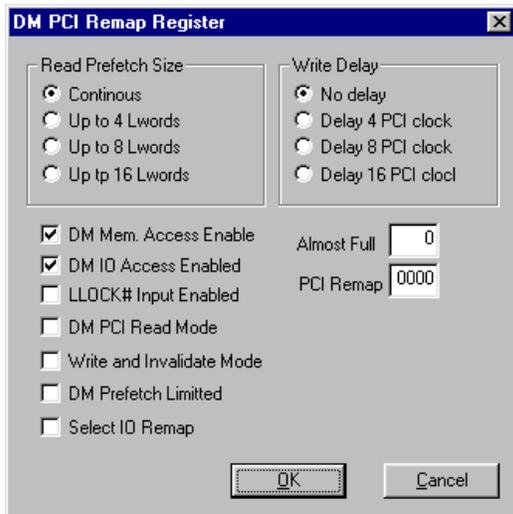


Figure 6-6. Direct Master PCI Remap Dialog Box

The Direct Master PCI Remap Dialog Box

The Direct Master (DM) PCI Remap dialog box provides information on the current value of the PCI Base Address (Remap) Register for Direct Master to PCI Memory and allows modification of that value (see Figure 6-6).

The DM Configuration I/O Address Dialog Box

The DM Configuration I/O Address dialog box provides information on the PCI configuration Address Register for Direct Master configuration and allows modification of that value (see Figure 6-7).

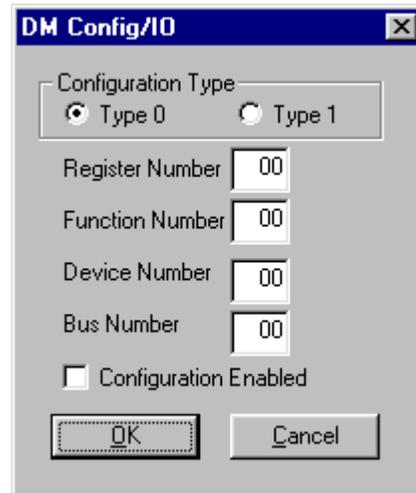


Figure 6-7. Direct Master Configure Dialog Box

The Region 1 Dialog Box

The Region 1 dialog box provides information on the Local Address Space 1 Bus Region Descriptor register and allows modification of that value (see Figure 6-8).

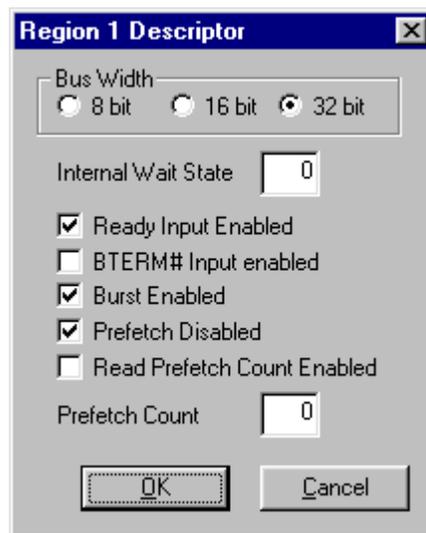


Figure 6-8. Region 1 Descriptor Dialog Box

6.1.3 The Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays the current register values. All register values can be modified by changing the contents of any register.

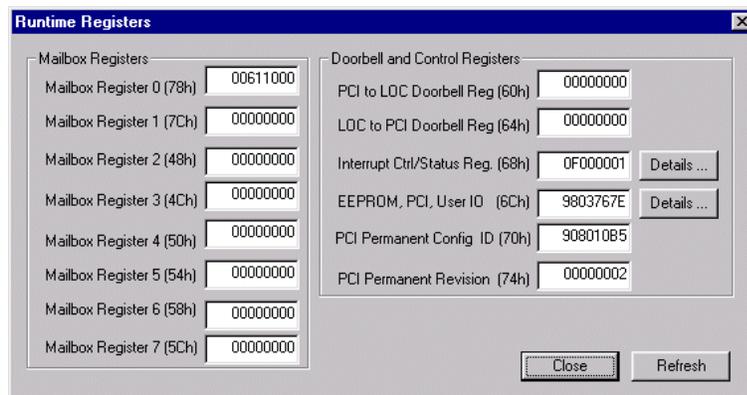


Figure 6-9. Runtime Registers Dialog Box

The Interrupt Control/Status Register Dialog Box

The Interrupt Control/Status Register Dialog Box provides information on the current value of the Interrupt Control/Status register.

The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

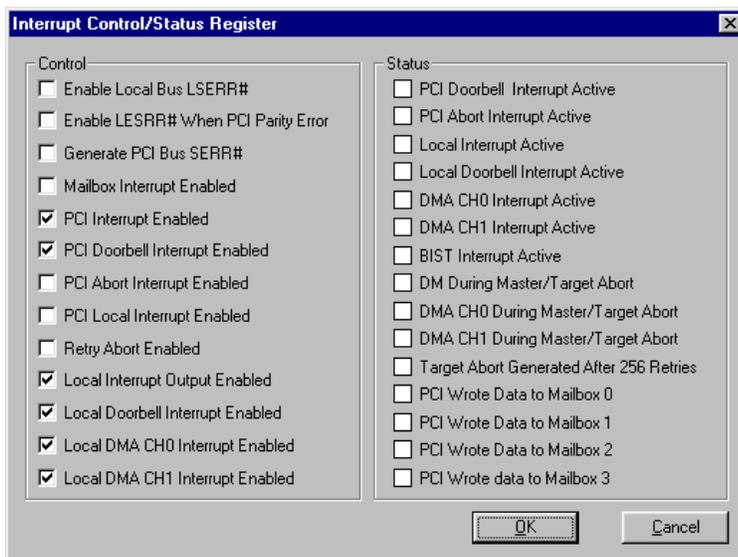


Figure 6-10. Interrupt Control and Status Dialog Box

The EEPROM, PCI, User IO Dialog Box

The EEPROM, PCI, User IO dialog box provides information on the current contents of the EEPROM Control, PCI Command Codes, User I/O Control, Initialization Control Register and allows modification of that value (see Figure 6-11). The Status section contained in this dialog box contains values that cannot be modified.

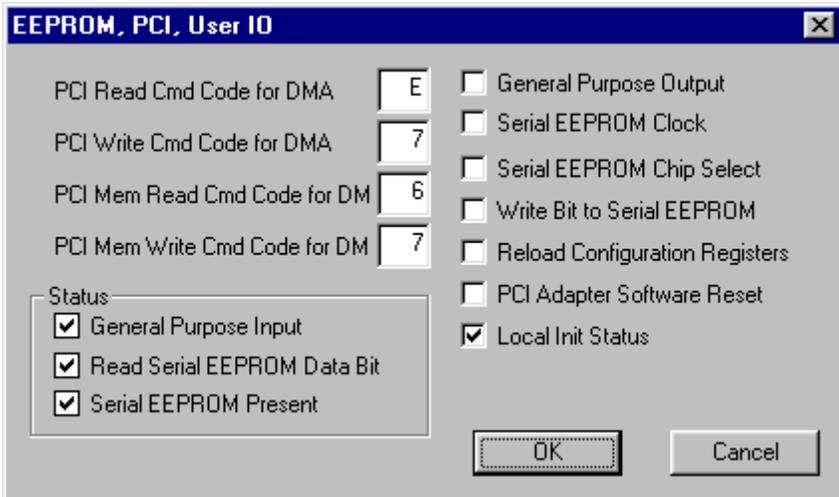


Figure 6-11. EEPROM, PCI, User IO Details Dialog Box

6.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for both DMA channels (see Figure 6-12).

The Start and Abort Transfer buttons, in Figure C-12, initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Channel Enable bit, enables DMA transfers and activates the Start, Abort, and Clear Interrupt buttons.

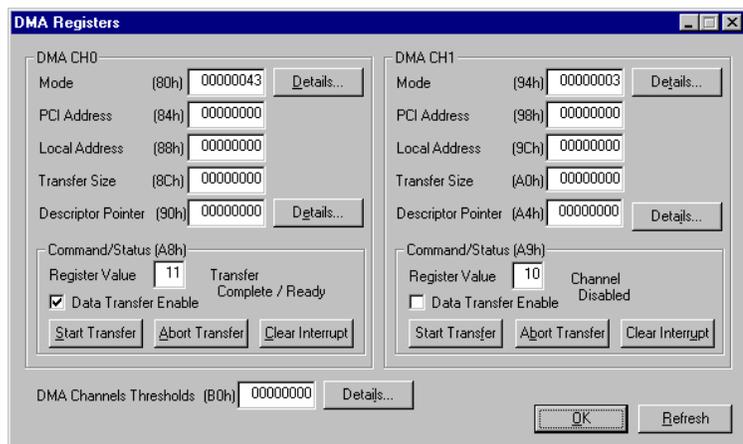


Figure 6-12. DMA Registers Dialog Box

The DMA Mode Dialog Box

The DMA Mode dialog box provides information on the current value the DMA Channel's Mode Register and allows modification of that value (see Figure 6-13).

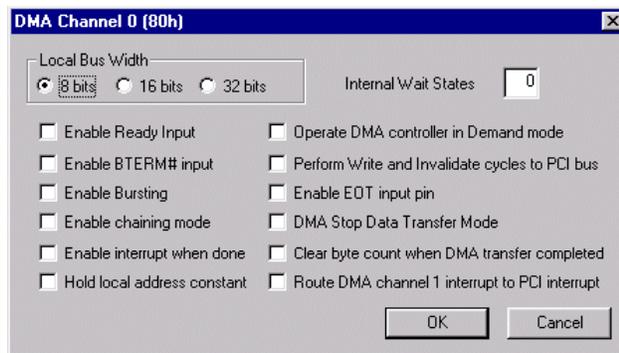


Figure 6-13. DMA Mode Dialog Box



Figure 6-14. Descriptor Pointer Dialog Box

The Descriptor Pointer Dialog Box

The Descriptor Pointer dialog box provides information on the current value of the DMA Channel's Descriptor Pointer Register and allows modification of that value (see Figure 6-14).

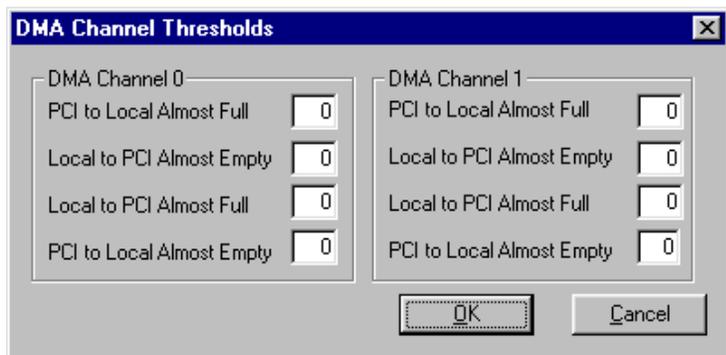


Figure 6-15. DMA Channels Threshold Dialog Box

The DMA Channels Threshold Dialog Box

The DMA Channels Threshold dialog box provides information on the current value of the DMA Threshold Register and allows modification of that value (see Figure 6-15).

6.1.5 The Messaging FIFO Register Group Dialog Box

The Messaging FIFO Register Group dialog box contains the current values for the Messaging FIFO Registers as shown in Figure 6-16.

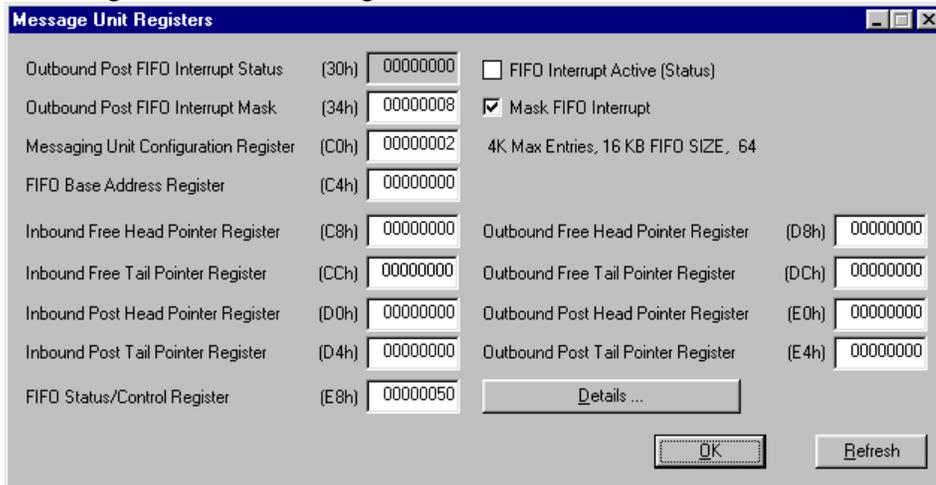


Figure 6-16. Messaging Unit Registers Dialog box

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified.

The FIFO Status/Control Register Dialog Box

The FIFO Status/Control Register dialog box provides information on the current value of the Queue Status/Control Register and allows modification of that value (see Figure 6-17).

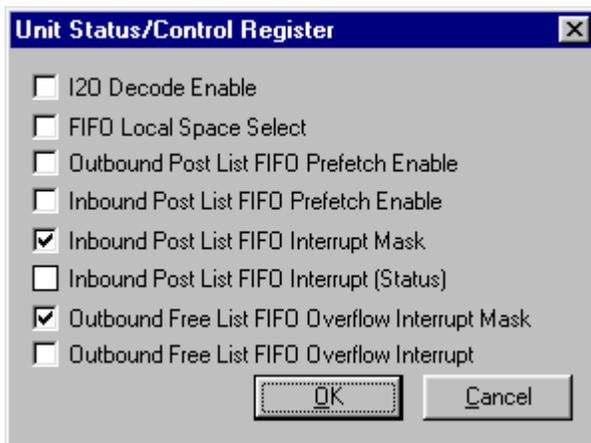


Figure 6-17. FIFO Status/Control Register Dialog Box



This page intentionally left blank.

Appendix A. Troubleshooting

In this section you can find solutions to common problems found with PLXMon 99. If a problem encountered is not listed here, please contact PLX customer support (Section 1.4)

I know I have a RDK in my computer, yet when I start PLXMon 99, the program will only give me serial access. (WinNT only)

This means the driver was unable to find a “supported” device on your computer. When this happens, the driver will unload itself. Use the event viewer to verify this occurred and to check the cause. Use the driver wizard to add the vendor and device ID of your PCI device to the supported list. Then either restart the computer or manually restart the driver. Instructions for adding a supported device can be found in the SDK User’s Manual.

After installing my custom board (the Vendor and Device IDs are my own) the Add New Hardware Wizard in Windows98 cannot find my board.

-or-

When adding two different RDK boards at the same time the Add New Hardware Wizard cannot differentiate between them. How do I know which board is which?

The “Add New Hardware Wizard” in Win98 relies on the Vendor and Device IDs of the PCI cards you are inserting. If a custom board is inserted, you must tell the Wizard that the .inf (installation script file) is located in the Inf directory under the Windows system directory. Then select “Unknown PCI XXXX board” depending on the PLX chip that is present on the RDK. The Inf directory is hidden, so make sure you “View all types” within the viewing options of Explorer to find it. Be sure to add new RDKs once at a time to avoid confusing the Wizard.



This page intentionally left blank.

Appendix B. Glossary Of Terms

Back End Monitor (BEM) or (BEM L1)

The Back End Monitor is an embedded program that can be compiled into the embedded software running on a PLX RDK. Its purpose is to scan the serial input and steal any data that it determines to be a BEM command. The BEM commands allow reads, writes, and resets of a PLX RDK. For more information about BEM, see the PLX SDK User's Manual.

COFF File Format

Coff files normally are the final data format for a RAM application compiled for use on the PCI and cPCI 9054RDK-860 RDK boards. The data contained within this file is Big-Endian.

IBM-401B Image File

IOP RAM programs compiled for the PCI 9080RDK-401B for RAM will be created in this format. The data is stored in Big-Endian format.

IOP (Input / Output Platform)

This term is interchangeable with the Embedded platform or Local side. This can mean all the software and/or hardware that is on a PLX RDK.

Motorola SRecord

This file format is produced as an intermediate file when compiling code for the PCI and cPCI 9054-RDK860. Data is not stored in any particular Endian format.

PCI bus

The PCI bus physically is the location (along with a slot) where the PLX RDK is inserted. The PCI bus can also be given as an address range with data accessible according to the PCI specification.

PCI SDK 3.0

This is the current version of PLX's PCI Software Development Kit Version 3.0

PLXMon 99

This application.



This page intentionally left blank.