



A. Product Status

The scope of this document encompasses the PCI SDK v3.0

Product status	Description	Production
PCI SDK v3.0	Production release	September, 1999

B. PCI SDK v3.0 Release Notes

1. Full Support for the following PLX Reference Design Boards:
 - IOP 480RDK;
 - CompactPCI 9054RDK-860;
 - PCI 9054RDK-860;
 - PCI 9054RDK-Lite;
 - PCI 9080RDK-401B;
 - PCI 9080RDK-860;
 - PCI 9080RDK-SH3;
 - PCI 9080RDK-RC32364.
2. Advanced Local (IOP) APIs for PLX's IOP 480, PCI 9054 and PCI 9080 devices;
3. Debug and Release versions of the Local (IOP) APIs;
4. Local (IOP) APIs compiled with DIAB compiler in ELF and COFF format;
5. Board Support Package for PLX reference design boards;
6. Back-End Monitor application debugger;
7. VxWorks (RTOS) support for PCI9054RDK-860 and CompactPCI 9054RDK-860;
8. Local (IOP) Direct Memory Access (DMA) Resource Manager that supports three modes of operation: Block DMA Mode, Scatter/Gather DMA Mode, Shuttle DMA Mode;
9. Graphical User Interface (GUI) screens that are based on PLX device registers;
10. Compatible with the IOP 480, PCI 9054 and PCI 9080 chips;
11. EEPROM Edit Utility to program a blank EEPROM;
12. Split Screen Interface, allowing command line input while receiving serial data;
13. Serial communications with an IOP's debug port. This feature is compatible with PLX's Back-End Monitor protocol.
14. A built-in downloader providing support for the following image standards: Motorola S-Record, IBM-ELF Image Files, COFF, and Binary. This feature supports downloading to RAM and FLASH devices through the PCI bus and the serial port.
15. PLX EEPROM Configuration screens allow modification of the contents of NM93CS46, NM93CS56, and NM93CS66 EEPROMs. These configuration screens allow you to load and save values from and to a file.
16. Customizable Hot-Links: this feature allows users to launch Win32 compatible programs such as custom tests and sample programs.
17. PLXMon 99 has an integrated downloader application: allows users to download custom code to the target board.
18. Support for custom (OEM) board designs and customer vendor and device IDs.
19. PLXMon 99 supports PLX's RDKs (listed in item 1 above) in Serial Debug Mode.
20. Supports Unix Environment at source code level. (Refer Section C for detailed information).

C. Unix Environment Support

The PCI SDK v3.00 CD-R contains a directory called "unix". Files under this directory and its subdirectories were copied from the PCI SDK installation directory after each PCI SDK release is installed under Windows system.

Directory name	Files copied from the installed PCI SDK package
pcisdk300	PCI SDK v3.00 (September, 1999)
pcisdk211	PCI SDK 2.1 Rev. 1 (December, 1998)
pcisdk21	PCI SDK v2.1 (November, 1998)
pcisdk20	PCI SDK v2.0 (July, 1998)
pcisdk122	PCI SDK v1.2.2 (March, 1998)

Note:

All the directories and files were created under Windows where there is no difference between filenames in lowercase and FILENAMES IN UPPERCASE, therefore some effort has to be made to make the directory names consistent before re-compiling of the source code can be done under Unix or Linux systems.

D. Updates to PCI SDK User's Manual

This section lists the updates made to the PCI SDK User's Manual after printing was finished. The CD version contains the updated manual. It can be located in the "<INSTALLPATH>\Documentation" folder.

D.1 Page iv, The List of Tables should read:

List of Tables

TABLE 3-1. BEM COMMANDS	3-13
TABLE 5-1. BASIC INFORMATION ABOUT IOP 480RDK	5-1
TABLE 5-2. BASIC INFORMATION ABOUT PCI 9054RDK-860	5-2
TABLE 5-3. BASIC INFORMATION ABOUT COMPACTPCI 9054RDK-860	5-4
TABLE 5-4. BASIC INFORMATION ABOUT PCI 9080RDK-401B.....	5-5
TABLE 5-5. BASIC INFORMATION ABOUT PCI 9080RDK-860	5-7
TABLE 5-6. BASIC INFORMATION ABOUT PCI 9080RDK-SH3	5-8
TABLE 5-7. BASIC INFORMATION ABOUT PCI 9080RDK-RC32364	5-10

D.2 Page 1-3, Section 1.8 should read:

1.8 Development Tools

Development tools used to develop the PCI SDK include:

- Win32 Applications: Microsoft Visual C++ 5.0, with Microsoft Developer Studio;
- Win32 Applications: Microsoft Platform Software Development Kit (SDK);
- Windows NT 4.0 Drivers: Microsoft Windows NT 4.0 Device Driver Kit (DDK);

- Windows 98 Drivers: Microsoft Windows 98 Device Driver Kit (DDK);
- IOP 480RDK IOP Software:
 1. IBM High C/C++ PowerPC Cross-Compiler, version 1.0 (7/31/96); and IBM 401 EVB Software Support Package, version 1.6.4 (4/1/97)
 2. DIAB Data, Inc. Compiler and Linker for PowerPC, version 4.0b and version 4.3p6;
- PCI 9080RDK-860, CompactPCI and PCI 9054RDK-860 IOP Software: DIAB Data, Inc. Compiler and Linker for the PowerPC, version 4.0b and version 4.3p6;
- PCI 9080RDK-401B Software:
 1. IBM High C/C++ PowerPC Cross-Compiler, version 1.0 (7/31/96); and IBM 401 EVB Software Support Package, version 1.6.4 (4/1/97)
 2. DIAB Data, Inc. Compiler and Linker for PowerPC, version 4.0b and version 4.3p6;
- PCI 9080RDK-RC32364 IOP Software: IDT/c Cross Compiler System Version 5.5/7.0 GNU Developer's Kit.
- PCI 9080RDK-SH3 IOP Software: Cygnus GNU compiler: gcc version 2.7-96q3a.

D.3 Page 3-8, Section 3.4.2 should read:

3.4.2 IOP API Library

The IOP API library contains the code for all the documented API functions. This code is standard for all IOP applications and is independent of the board configuration. The code directly calls the PLX chip (no intermediary functions).

There are at least two IOP APIs for each PCI device: Release and Debug. Both libraries are the same except the release version eliminates many of the parameter validation steps that are performed in the debug version, and hence performance is increased when using the release version of the API. All debug libraries contain a 'd' suffix in their name (E.G. `api860d.a`). Release libraries do not contain the 'd' suffix in their name (E.G. `api860.a`).

For the PowerPC CPU type supported by DIAB compiler, two sets of release and debug version IOP API libraries are created: Release and Debug versions in ELF format; Release and Debug versions in COFF format. Libraries in ELF or in COFF format exist in DIAB-ELF or DIAB-COFF subdirectory respectively under the directory `<INSTALLPATH>\Iop\Lib\RDK_NAME`.

RDK_NAME is one of the following five RDK directory names.

- IOP 480RDK;
- 9054RDK-860;
- CPCI9054RDK-860;
- PCI 9054RDK-Lite;
- 9080RDK-860; and
- 9080RDK-401B.

For the 9080RDK-401B and IOP 480 RDK, one more set of release version and debug version IOP API libraries are created with the IBM High C/C++ PowerPC Cross-Compiler. Libraries created by

IBM High C/C++ PowerPC Cross-Compiler are located under the
 <INSTALLPATH>\Iop\Lib\RDK_NAME\IBM-ELF directory where **RDK_NAME** can be either
 9080RDK-401B or IOP480RDK.

Note: Each PLX chip has its own IOP API library specifically designed to complement its features. To implement more than one PLX chip on one board, a new library must be created. This library would combine the features of each chip and have new functions to accent the features achieved by grouping the PLX chips.

D.4 Page 3-13, Table Caption in Section 3.4.3. should read:

Table 3-1. BEM Commands

D.5 Page 5-2, the Figure 5-1 should read:

IOP480 EEPROM Values

PCI Configuration Registers:

Device ID (D4h)	0480	Vendor ID (D6h)	10B5	Class Code (D8h)	06800001	PM Scale (ECh)	00000000
Subsystem ID (DC h)	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)	FFF00000	(4Ch)	00000000	(70h)	FFF80000	(94h)	00000000	(B8h)	0000C027
(08h)	00000000	(2Ch)	00000000	(50h)	50000000	(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)	FE000000	(3Ch)	00000000	(60h)	00000000	(84h)	FFF00000	(A8h)	00000000	(CCh)	00000000
(1Ch)	00000000	(40h)	F0000000	(64h)	0007310E	(88h)	00000000	(ACh)	FFF00000	(D0h)	00000000
(20h)	FE000000	(44h)	40000000	(68h)	00020178	(8Ch)	00000000	(B0h)	00000102	(FCh)	80004000

IOP480 Clock Frequency:

Clock Frequency (MHz) (100h)

OK Cancel Apply Load File Save As...

Show Offset in:

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

E. Updates to PCI SDK Programmer's Reference Manual

This section lists the updates made to the PCI SDK Programmer's Reference Manual after printing was finished. The CD version contains the updated manual. It can be located in the "<INSTALLPATH>\Documentation" folder.

E.1 Page 2-9, The second paragraph should read:

Once the DIAB PowerPC compiler compiles all the necessary object files and links the object files with necessary libraries into an executable image file in COFF format. The executable image file is converted from COFF format into Motorola S-Record format, which is in turn converted into a pure binary image file by `s2bin.exe`, a utility program provided in the PCI SDK package. The binary image generated by `s2bin.exe` is further processed by `PpcBinFix.exe`, another utility program provided in the PCI SDK package, so that a branch opcode is inserted at the very end of the binary image. And the branch opcode inserted by `PpcBinFix.exe` branches off to the very beginning of the image where another branch opcode MUST exist.

E.2 All 480BinFix.exe should read PpcBinFix.exe in following pages:

- Page 2-9;
- Page 2-11;
- Page 2-12; and
- Page 2-14;

PlxSpuBaudRateSet

Syntax:

```
RETURN_CODE PlxSpuBaudRateSet(IN BUS_INDEX busIndex,  
                               IN U32 baudRate,  
                               IN U32 lclkFreq);
```

PLX Chip Support:

IOP 480

Description:

Sets baud rate for SPU operation. Setting baud rate can also be done using PlxSpuInit.

- *busIndex* is the bus index,
- *baudRate* is the baud rate to be operated on by SPU. It should be one of the following value, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, and
- *lclkFreq* is the external system clock (LCLK) frequency in Hz and used as the Baud Rate Generator Clock input and must be exactly the same as the frequency indicated by hardware.

Return Value:

Return Value	Description	API Support
ApiSuccess	The function returned successfully.	IOP 480
ApiFailed	The API call fails.	IOP 480
ApiInvalidBusIndex	The <i>busIndex</i> value is invalid	IOP 480

Usage:

```
RETURN_CODE    rc;  
U32 baudRate;  
  
baudRate = 38400;  
  
/* Initialize the Serial Port Unit */  
rc = PlxSpuBaudRateSet(PrimaryPciBus, baudRate, 66 * 1000 * 1000);  
  
if (rc != ApiSuccess)  
    /* error handler code goes here */
```

E.4 Page 3-138, The entire page should read:

PlxSpuInit

Syntax:

```
RETURN_CODE PlxSpuInit(IN BUS_INDEX busIndex,  
                      IN PSPU_DESC pSpuDesc);
```

PLX Chip Support:

IOP 480

Description:

Initializes the Serial Port Unit of the IOP 480

- *busIndex* is the bus index,
- *pSpuDesc* is a pointer to the parameters that will be used to initialize the SPU.

Return Value:

Return Value	Description	API Support
ApiSuccess	The function returned successfully.	IOP 480
ApiFailed	The initialization failed	IOP 480
ApiNullParam	The <i>pSpuDesc</i> parameter is invalid.	IOP 480
ApiInvalidBusIndex	The <i>busIndex</i> given is invalid	IOP 480

Usage:

```
RETURN_CODE rc;  
SPU_DESC spuDesc;  
memset(&spuDesc, 0, sizeof(SPU_DESC));  
  
spuDesc.LclkFreq = 66 * 1000 * 1000; /* LCLK frequency in Hz */  
spuDesc.BaudRate = 38400;           /* Baud Rate */  
spuDesc.DTR = 1;                    /* DTR signal is active */  
spuDesc.DB = 1;                      /* Data bits is 8 */  
spuDesc.SB = 1;                      /* two stop bits */  
rc = PlxSpuInit(PrimaryPciBus, &spuDesc); /* Init. the SPU */  
if (rc != ApiSuccess)  
    /* error handler code goes here */;
```

Cross Reference:

Referenced Item	Page
PSPU_DESC	5-64

Serial Port Descriptor Structure

```
typedef struct _SPU_DESC
{
    U32 BaudRate;
    U32 LclkFreq;
    U32 LM           : 2;
    U32 DTR         : 1;
    U32 RTS         : 1;
    U32 DB          : 1;
    U32 PE          : 1;
    U32 PTY         : 1;
    U32 SB          : 1;
    U32 Reserved    : 24;
} SPU_DESC, *PSPU_DESC;
```

Affected Register Location (offset from SPU base address)

Structure Element	PCI 9080	PCI 9054	IOP 480
BaudRate	N/A	N/A	0x4000 0010, 0x4000 0014
LM	N/A	N/A	0x4000 0018, 0-1
DTR	N/A	N/A	0x4000 0018, 2
RTS	N/A	N/A	0x4000 0018, 3
DB	N/A	N/A	0x4000 0018, 4
PE	N/A	N/A	0x4000 0018, 5
PTY	N/A	N/A	0x4000 0018, 6
SB	N/A	N/A	0x4000 0018, 7

Purpose

Data type used to initialize the Serial Port Control.

Members

BaudRate

Specify the baud rate at which the Serial Port Unit (SPU) operates. They should be one of the following value, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

LclkFreq

Specify the external system clock (LCLK) frequency in Hz. The LCLK is the Baud Rate Generator Clock input.

LM

Loopback Modes.
 00 - Normal mode
 01 - Internal Loopback mode
 10 - Automatic Echo mode
 11 - Reserved

E.6 Page 5-64: The entire page should read:

DTR

Data Terminal

0 - DTR signal is inactive

1 - DTR signal is active

RTS

Request to Send.

0 - RTS signal is inactive

1 - RTS signal is active

DB

Data Bits.

0 - 7 Data bits

1 - 8 Data bits

PE

Parity Enable

PTY

Parity.

0 - Even parity.

1 - Odd parity.

SB

Stop bits.

0 - One stop bit

1 - Two stop bit

Copyright © 1999 by PLX Technology, Inc. All rights reserved. PLX is a trademark of PLX Technology, Inc. which may be registered in some jurisdictions.

All other product names that appear in this material are for identification purposes only and are acknowledged to be trademarks or registered trademarks of their respective companies.

Information supplied by PLX is believed to be accurate and reliable, but PLX Technology, Inc. assumes no responsibility for any errors that may appear in this material. PLX Technology reserves the right, without notice, to make changes in product design or specification.