PCI Bridge Compatibility

DEFAULT MEMORY SPACE

The default memory location of some boards is in the lowest 1M byte of PCI memory space. If a board must operate in any other memory space (ie, 32-bit or 64-bit address), a customer can specify the memory location by adding -M0 (32-bit) or -M4 (64-bit) at the end of the model numbeer.

MEMORY SPACE OPTIONS

In addition to options provided in the spec for each board (eg, FIFO depth, analog board options -AB which are shown in tables in the spec) a customer can also specify the default memory location of the board to select how the EEPROM (on the board) is programmed.

To specify the default memory location, simply add -Mn (where n= 0, 2, 4, 6, ash shown in the table below) to the end of the model number. For example: PMC-ADADIO-312-M0. The -M0 indicates that the board should be configured in 32-bit memory address space.

The ordering options available are:

Ordering Option	Value in LAS0RR
-M0	0xFFFFFFE0 = Locate anywhere in 32-bit PCI memory address space
-M2	0xFFFFFFE2 = Locate below 1M byte in PCI memory address space
-M4	0xFFFFFFE4 = Locate anywhere in 64-bit PCI memory address space
-M6	0xFFFFFFE6 = Reserved

In the event a board is received with a non-desirable option, a software routine which can be used to modify the EEPROM can be emailed. Otherwise a customer can return the board for modification of the EEPROM.

Problem Description

An access problem occurs when a board is located on the side of the PCI bridge furthest from the CPU (ie, downstream) and it is set to request memory space in the first M-byte space. In this case, it will not be possible for the board to be configured by Windows upon power-up; the result of this event is a Windows “PCI error 21”.

Problem Cause

The problem is caused when the board is located below 1 MB in PCI address space. The memory location of the board is set by the PLX configuration EEPROM when it initializes the

Local Address Space 0 Range Register (LAS0RR) for PCI to Local Bus (see Table 4-33 in PLX 9080 manual). The result, of that initialization, is that a request is generated to the operating system to relocate the board below 1MB in address space. This is usually OK if the board is plugged in directly to the PCI bus on a Windows PC, but will fail on most PC BIOSs because of the PCI bridge.

The bridge chip will be assigned to some high address apace, typically 0xFDFC0000. Since the board is requesting that it be assigned to an address below 1MB, an error will be returned by most BIOSs.

Solution

The solution is to change the PLX configuration or EEPROM contents for the **Local Address Space 0 Range Register for PCI to Local Bus.** The contents of this register are described in the following table (a partial excerpt of Table 4-33 in the PLX 9080 manual).

A utility routine (plxmon2.zip) can be can be obtained from the PLX web site www.plxtech.com/tools/software It should be enough to solve any customer configuration problems; it runs under both Windows and DOS. It is well documented and compact. It allows the user to change a board to 32-bit address space.

Table 4-33. (LAS0RR; PCI:00h, LOC:80h) Local Address Space 0 Range Register for PCI to Local Bus. Note that the PCI address of this register (**LAS0RR)** is offset 00h from the base address that is loaded into the PCI configuration register for this board, ie, the **Base Address for Configuration Registers (PCIBAR0** is memory mapped, while **PCIBAR1** is I/O mapped**).**

Bit 0, R/W	Memory space indicator. Value of 0 indicates Local address space 0 maps into PCI memory space. Value of 1 indicates address space 0 maps into PCI I/O space.
Bits 2,1 R/W	If mapped into memory space, encoding is as follows: 00 = Locate anywhere in 32 bit PCI address space 01 = Locate below 1 MB in PCI address space 10 = Locate anywhere in 64 bit PCI address space 11 = Reserved If mapped into I/O space, bit 1 must be set to 0. Bit 2 is included with bits [31:3] to indicate decoding range.

Although the serial EEPROM can be changed by a software routine (see **Procedure for Changing the Serial EEPROM,** below), the suggested solution is to change two configuration registers during the initialization of the board. This involves writing to two registers: the **Local Address Space 0 Range Register for PCI to Local Bus (LAS0RR),** and the **PCI Base Address 2 for Local Address Space 0** (**PCIBAR2).**

The problem will typically occur when the value in the **Local Address Space 0 Range Register for PCI to Local Bus** is set to 0xFFFFFFE2. So, typically the solution will be to write 0xFFFFFFE0 to this register.

Values for the **Local Address Space 0 Range Register**

The actual values that can be written into the **Local Address Space 0 Range Register for PCI to Local Bus (LAS0RR)** are configurable at the factory (or by the user):

Ordering Option	Value in LAS0RR
-M0	**0xFFFFFFE0 = Locate anywhere in 32 bit PCI address space**
-M2	0xFFFFFFE2 = Locate below 1 MB in PCI address space
-M4	0xFFFFFFE4 = Locate anywhere in 64 bit PCI address space
-M6	0xFFFFFFE6 = Reserved

Once the **LAS0RR** has been modified, then the **PCIBAR2** must also be initialized.
PCIBAR2 is the **PCI Base Address 2 for Local Address Space 0;** it is a PCI config register located at offset 18h.

Values for the **PCI Base Address 2 for Local Address Space 0**

If the value in **LAS0RR** is changed after the "Plug-and-Play" initialization has been completed, then the value in **PCIBAR2** must be loaded with a valid address, ie, a 32-bit address that places the Local Configuration Registers at a valid location in PCI address space.

However, if the EEPROM itself is modified and the system is re-booted, then PCIBAR2 will be automatically loaded by the "Plug-and-Play" initialization software.

A valid location in PCI address space should be obtainable by calling an address space allocation routine.

Procedure for Changing the Serial EEPROM

The serial EEPROM contents for the **Local Address Space 0 Range Register for PCI to Local Bus** can be changed via the **Serial EEPROM Control Register** (a runtime control register located at offset 6Ch from the PCI Base Address). The contents of this register are described in the following table (a partial excerpt of Table 4-33 in the PLX 9080 manual). The procedure for changing the serial EEPROM is cumbersome; however, a DOS-based routine, which accomplishes this, is available to our customers free of charge. Also, the board can be returned for reprogramming; however, this is not the preferred method since it would require a different configuration to be shipped for different customers and different applications. GSC will, if requested, ship any board with the LAS0RR initialized to any value specified in the model number (see the ordering options in the table above, ie, -M0, -M2, -M4, and -M6).

Table 4-59. (CNTRL; PCI:6Ch, LOC:ECh) Serial EEPROM Control, PCI Command Codes, User I/O Control, & Init Control Register for PCI to Local Bus. Note that the PCI address of this register (**CNTRL)** is offset 6Ch from the PCI base address for this board. Bits 0 to 23 and bits 28 to 31 are skipped in this excerpt since they are not relevant.

Bit 24, R/W Serial EEPROM Clock for Local or PCI Bus Reads or Writes to serial EEPROM. Toggling this bit generates a serial EEPROM clock. (Refer to the manufacturer's data sheet for the particular serial EEPROM being used on this board).

Bits 25 R/W Serial EEPROM Chip Select. For local or PCI bus reads or writes to serial EEEPROM, setting this bit to 1 provides the serial EEPROM chip select.

Bits 26 RO Write Data Bit to serial EEPROM. For writes, this bit is the input to the serial EEPROM. This data bit is clocked into the serial EEPROM by the serial EEPROM clock.

Bits 27 R/W Read serial EEPROM Data Bit. For reads, this input bit is the output of the serial EEPROM. This data bit is clocked out of the serial EEPROM by the serial EEPROM clock.

Once the serial EEPROM has been modified, the "Plug-and-Play" initialization software will automatically initialize the **PCI Base Address 2 for Local Address Space 0** (**PCIBAR2)** upon power-up initialization.

The PMC-OPTO32 is Shipped in PCI I/O Space

The EEPROM on the PMC-OPTO32 "pre-configures" (initializes) the PMC-OPTO32 local registers to operate in PCI I/O space and the software driver properly sets up and uses I/O space transactions for PMC-OPTO32 register accesses.

However, this can be changed by the user to Memory space in either the PLX-PCI9060ES or PLX-9080 chips, if need be. One of the differences between the PLX-PCI9080 chip and the PLX-PCI9060ES chip is that the 9080 chip has added another PCI configuration base address offset register for Local Space 1. What this means is that both the PLX-PCI9080 local registers and the PMC-OPTO32 registers can operate simultaneously in I/O space and Memory Space, given that the end user configures the base address information properly in the PCI Configuration registers. At the moment, I can't think of a reason why the end user might want to do this, but the flexibility to do so is there. This was not possible using the PLX-PCI9060ES chip.

As to whether I/O space is a better choice or not, I guess it depends on the application. Some processors aren't capable of performing I/O space transactions, so memory space is the only choice. Also, I/O space is typically allocated less address space (densely populated) than memory space. However, the choice really depends on the total functionality of the PCI device and the processor that it is used with. Another limitation of I/O space is that I/O space addresses cannot be write posted or pre-fetched, whereas Memory space addresses can.