

TAMS 80100B PCI-VXI Controller



Installation & Operations Instructions

Copyright © Test & Measurement Systems Inc. 2000 - 2004

Installation & Operations Instructions for the TAMS 80100B PCI-VXI Controller

HP-UX Workstation and Linux Versions

Test & Measurement Systems Inc.
750 14th Street SW
Loveland, CO 80537
USA

Telephone (970) 669 6553
Fax (970) 669 3090
Web Site www.tamsinc.com

Contents

- FEATURES 4
- KNOWN PROBLEMS..... 4
- INSTALLING THE TAMS 80100B PCI-VXI CONTROLLER 5**
 - INSTALLING THE PCI CABLE DRIVER CARD 5
 - INSTALLING THE VXI CONTROLLER CARD 5
 - EXTERNAL CLOCK ROUTING 5
 - SELECTING THE INTERNAL ADDRESS MODIFIER 5
 - SOFTWARE INSTALLATION..... 6
 - HP-UX 10.20 6
 - HP-UX 11i 7
 - LINUX 7
- SOFTWARE CONFIGURATION 8**
 - TAMS 80100 PARAMETERS 8
 - CONFIGURING WITH IOSETUP. 10
 - HP-UX 10.20 AND 11i..... 10
 - LINUX 10
 - CONFIGURING WITH HWCONFIG.CF AND SICLCONF 13
 - ON HP-UX 10.20 AND 11i 13
 - ON LINUX 13
 - EDITING THE /ETC/OPT/SICL/IPROC.CF FILE..... 14
 - ON HP-UX 10.20 AND 11i 14
 - ON LINUX 14
- USING THE TAMS 80100B PCI-VXI CARD 15**
 - PROGRAMMING DIFFERENCES 15
 - LINUX SPECIFIC ISSUES 15
 - POWER-CYCLING CONSIDERATIONS 19
 - MULTI-CAGE CONFIGURATION 19

FEATURES

- A16, A24, A32; D08(EO), D16, D32 transfers
- High speed DMA transfers via the `iblockcopy()` and `iblockmove()` commands
- 16 bit IACK response cycles

KNOWN PROBLEMS

- MXI-1 access. An apparent VME bus violation on the part of the MXI-1 card currently prevent the TAMS 80100 card from accessing MXI-1 cards. TAMS has tested MXI-2 cage to cage and it appears to work without problems.

INSTALLING THE TAMS 80100B PCI-VXI CONTROLLER

INSTALLING THE PCI CABLE DRIVER CARD

The PCI Cable Driver Card can be installed in the host workstation or PC in any of the available 33MHz slots. The card works in both 3.3 and 5 volt slots and both 32 and 64bit connectors.

It is possible to install 33MHz PCI cards in a 66MHz PCI bus, but such an installation will slow the bus clock down to 33MHz.

The standard PCI card is labeled “33 Mhz, 2M Cable”. The 33 MHz refers to the on-board oscillator, which provides the clock for the cable between the two cards of the 80100B set. A cable of 2 meters length is used with the cardset.

A production change for the PCI card uses a newer bridge chip. These cards are labeled “33 MHz, 2M Cable, 21555”. Cards like this require a minimum revision of software of 2.5 for HP-UX 11.11, 1.5 for HP-UX 10.20, or 2.2 for Linux. Without these minimum revisions of software, the cards will not be recognized.

Jumper JP1, if installed, allows the clock received from the PCI bus (normally 33 MHz) to be used instead of the clock generated onboard. On some PC's, this will provide better system throughput. However, some systems may provide a dithered clock on PCI, which is not useable with the 80100B product. In those cases, the jumper must be removed.

NOTE Systems that are known to use a dithered PCI clock are the Bxxxx and Cxxxx workstations from HP. Other HP-UX workstations or PC's may use a dithered PCI clock, but none have been identified at the present time. If you believe that this may be a problem on your system, be sure that the jumper JP1 is removed.

INSTALLING THE VXI CONTROLLER CARD

EXTERNAL CLOCK ROUTING

As a Slot 0 controller, the 80100 PCI-VXI controller can supply the 10MHz reference clock, **CLK10**, either from it's own internal oscillator or from an external reference supplied through the front panel.

<u>Switch Position</u>	<u>Reference Clock Selected</u>
Internal (Down)	Internal Clock (default)
External (Up)	External Reference

When the external oscillator is selected, the **EXT CLK** green LED is illuminated on the front panel.

The 80100 PCI-VXI controller does not supply **CLK10** if it is not in Slot 0.

SELECTING THE INTERNAL ADDRESS MODIFIER

During initialization, the VXI Controller card uses a VME user defined Address Modifier to access it's internal registers. For a given system, this Address Modifier should be reserved for the exclusive use of the VXI Controller card.

It is unusual for VXI cards to use user defined Address Modifiers, under normal circumstances the Internal Address Modifier should not have to be changed from the default of 0x10 (rotary DIP switch position 0).

In the event that the VXI Controller card is used in a system where there is a conflict with another card using the same Address Modifier, the Address Modifier of the 80100 may be moved by changing the position of the rotary DIP switch on the VXI Controller card. The address modifier will be 0x1x, where x is the hexadecimal value of the rotary DIP switch setting.

If the DIP switch position is changed, this change must be reflected in the “**DIP Switch**” parameter for the software configuration of the card.

SOFTWARE INSTALLATION

HP-UX 10.20

To use the TAMS 80100 PCI-VXI Controller with HP-UX 10.20, you will need to have installed the *TAMS N2091 I/O Libraries for HP-UX 10.20*. The 80100 driver is included as part of this version of the *I/O Libraries for HP-UX 10.20* (though not with Agilent's *E2091 I/O Libraries for HP-UX 10.20*), so no additional driver will need to be installed.

The *TAMS I/O Libraries* are virtually the same as those supplied by Agilent. They are simply extended to provide support for the 80100 PCI-VXI Controller card. The usage and directory structure are identical.

TAMS recommends using **swremove** to remove all previous versions of *TAMS I/O Libraries* and *Agilent I/O libraries* (product number E2091) before installing the most recent version of the *TAMS I/O Libraries*.

The *TAMS I/O Libraries* are supplied as a software depot (sd) package. Since TAMS uses a different version numbering scheme, one has to select the installation option in **swinstall** to allow a version with a lower number to be installed. To install:

1. Mount the supplied installation CDROM
2. As root, start **swinstall**: /usr/sbin/swinstall
3. In the options menu, select change options...
4. Select the “Allow installation of lower version than current” checkbox
5. Select “Ok”
6. Change the depot path to the CDROM mount point
7. Select the *TAMS I/O Libraries* (N2091) for installation

NOTE An error message may be logged in the swagent.log concerning a failure to copy a *vme1* file. This is of no significance.

HP-UX 11i

To use the TAMS 80100 PCI-VXI Controller with HP-UX 11i (11.11), you will need to have installed the *TAMS 72091 I/O Libraries for HP-UX 11i*. The 80100 driver is included as part of the *I/O Libraries for HP-UX 11i*, so no additional driver will need to be installed.

LINUX

To use the TAMS 80100 PCI-VXI Controller with Linux, you will need to have installed the *TAMS 82091 I/O Libraries for Linux*. The 80100 driver is included as part of the *I/O Libraries for Linux* (versions 2.0 and later), so no additional driver will need to be installed. Do note that only specific versions of the Linux kernel are supported by the Linux version of the *I/O Libraries*, so be sure to verify that you are using a supported kernel.

NOTE If the Linux kernel has had any additional patches applied, it is not supported, even if the kernel version number has not changed.

SOFTWARE CONFIGURATION

There are two methods of configuring an interface for use with SICL. One is using the *iosetup* graphical configuration tool. The other involves manually editing the *hwconfig.cf* file. No matter which method you choose, be certain that you have appropriate values for each of the parameters listed below before configuring your TAMS 80100 interface. Both of these methods also require that the *iprof.cf* file be edited.

TAMS 80100 PARAMETERS

Logical Unit Number	The SICL Logical Unit number of the interface. Each SICL interface card requires a unique logical address number. For RMB users, this is the same as the Select Code. 16 is the suggested address, since it is the default address that the VXI Resource Manager uses.
Symbolic Name	The SICL Symbolic Name for the interface. Each SICL interface card requires a unique symbolic name. 'vxi' is the suggested name, since it is the default name that the VXI Resource Manager uses.
PCI Slot Number	<p>The PCI slot number in the workstation or PC.</p> <p>On HP-UX, it is sometimes the case that correct slot information is not available. (For example, when using a PCI to PCI expander.) In these circumstances, for HP-UX systems with only a single PCI-VXI card, the slot number '0' can be used as a wildcard PCI slot address. This is not recommended in general practice.</p> <p>On Linux, the "slot number" is actually a composite of the PCI bus number and the PCI device number for the interface card. To get a list "PCI locations" where TAMS 80100 PCI cards are installed, run <code>/opt/sicl/bin/lst80100</code>, which is a utility included with <i>TAMS 82091 I/O Libraries for Linux</i>. On Linux, you can not use '0' as a wildcard PCI location.</p>
VXI Logical Address	The VXI logical address of the controller. This will nearly always be '0'. '0' is the required address of the VXI system controller.
VXI Slot Number	The VXI slot that the PCI-VXI Controller is located in. This is usually slot 0.
80100 Interrupt	The 80100 PCI-VXI card requires a VME interrupt for its internal use. It is suggested that a VME interrupt line be dedicated to the 80100 card. 7 is recommended, but this value can be changed to avoid conflicts with other VME or VXI cards. This interrupt cannot be disabled by the " VME Only Interrupts " parameter.
VME Only Interrupts	This parameter forces the TAMS 80100 to ignore specified VME interrupts.

In a normal VXI system, the Resource Manager assigns which device will be the interrupt handler for a specific VME interrupt line. In some systems, it may be necessary to override this behavior and force the TAMS 80100 to not respond to certain VME interrupt lines.

With the 80100 card this can be done by specifying a bit mask of the form: 0bxxxxxx, in which the left most bit disables VXI processing for VME IRQ 7 and the right most bit disables VXI processing for VME IRQ 1. The default is '0b0000000' which allows VXI interrupt processing for all VME interrupts.

Max Servant Addr The maximum servant address for which the 80100 is the commander. The 80100 is the commander for cards with logical address from 1 greater than the 80100's to the Max Servant Addr. The default is 255.

Shared Mem Space The 80100 has 2 Megabytes of on card memory that can be used as VXI shared memory. Valid settings are

- 0 No Shared Memory (default)
- 1 Memory Mapped to A24
- 2 Memory Mapped to A32

The VXI resource manager determines the actual location of the memory in the selected memory space.

VME BTO VME Bus timeout. The available values are

- | | |
|-------------|-------------------------------|
| 0, 5 or 256 | 256 microsecond BTO (default) |
| 1 or 16 | 16 microsecond BTO |
| 2 or 32 | 32 microsecond BTO |
| 3 or 64 | 64 microsecond BTO |
| 4 or 128 | 128 microsecond BTO |
| 6 or 512 | 512 microsecond BTO |

DIP Switch The rotary DIP switch setting. See the above section: *Selecting the Internal Address Modifier*. Normally, this is '0'.

Moveable Windows This parameter selects how many hardware windows are used in the A32 memory space. The 32MBytes of available memory space is divided as evenly as possible amongst the enabled hardware windows. The default value is 4 windows of 8MByte each. Valid values are 1 through 6.

CONFIGURING WITH *IOSETUP*.

HP-UX 10.20 AND 11i

The 80100 can be configured with the *iosetup* configuration program. Figure 1 shows the configuration dialog with a typical configuration. To use *iosetup* on HP-UX:

1. Start the *iosetup* program: `/opt/sicl/bin/iosetup`.
2. From the left hand menu select "TAMS 80100 VXI".
3. Enter the appropriate parameters as described in the section *TAMS 80100 Parameters*.
4. Select 'OK' in the 80100 Driver Window.
5. Select 'Done' in the Main Window.
6. Select 'OK' to accept the new configuration.
7. Select 'OK' to build the 80100 driver into the kernel.
8. Choose the 'reboot later' option.
9. Edit the *iprof.cf* file as described in the section: *Editing the /etc/opt/sicl/iprof.cf File*.
10. Reboot the workstation manually.

LINUX

Figure 2 shows the configuration dialog on Linux with the default configuration. On Linux, configure the 80100 using *iosetup* as follows:

1. Start the *iosetup* program: `/opt/sicl/bin/iosetup`.
2. From the left hand menu select "TAMS 80100 VXI Interface".
3. Enter the appropriate parameters as described in the section *TAMS 80100 Parameters*.
4. Select 'Ok' in the "TAMS 80100 VXI Interface" Window.
5. Select 'Ok' in the Main Window (or 'Configure VISA', if you are going to be using VISA).
6. The kernel driver will be built and installed.
7. Edit the *iprof.cf* file as described in the section: *Editing the /etc/opt/sicl/iprof.cf File*.
8. Restart the *sicl* service: `/sbin/service sicl stop ; /sbin/service sicl start`

NOTE You do not need to reboot the computer for the changes to take effect on Linux. Restarting the *sicl* service or running *iosetup* or *siclconf* causes the kernel driver to be reloaded with any configuration changes and also restarts *iprof*.

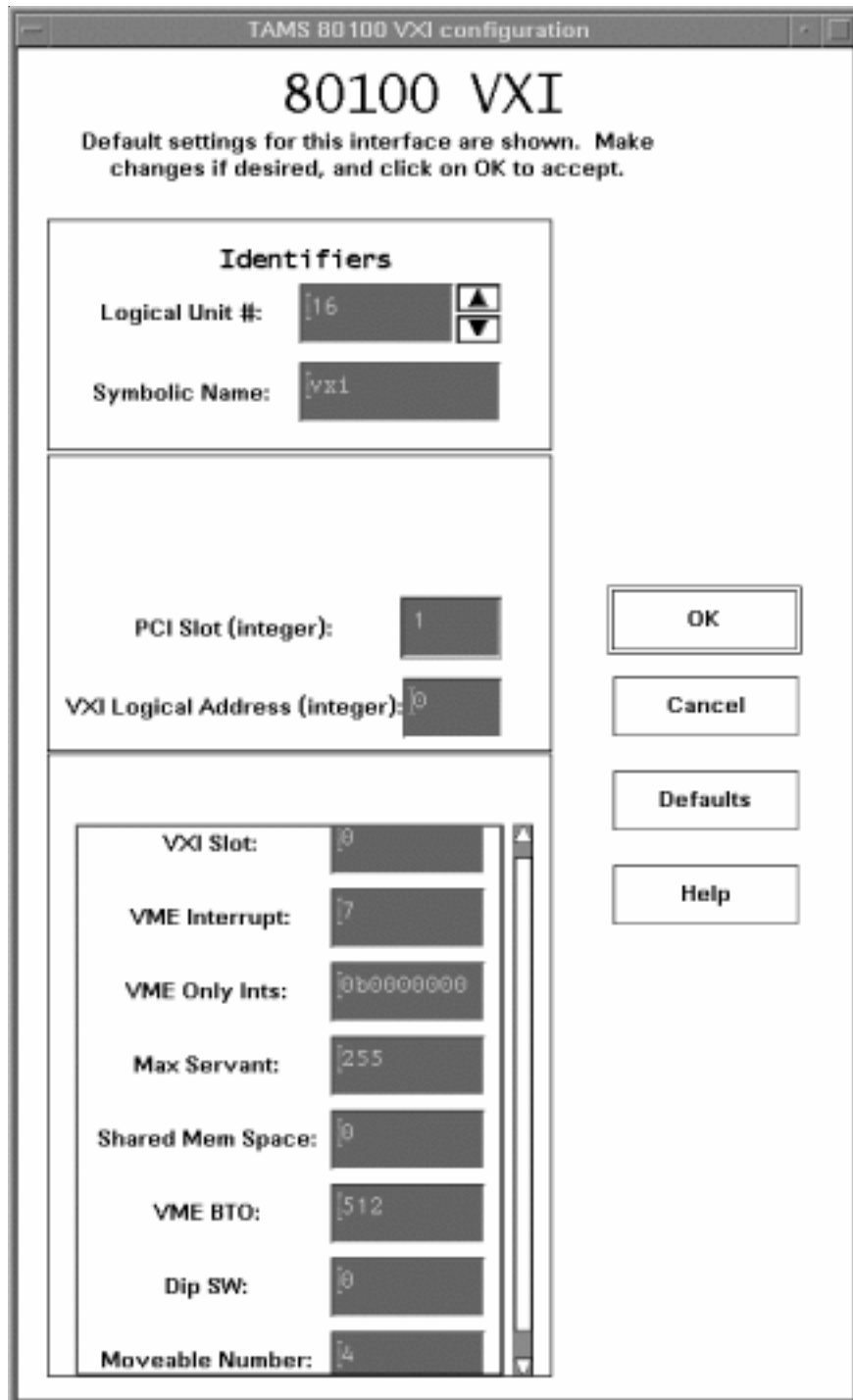


Figure 1. HP-UX 80100 configuration dialog with default values.

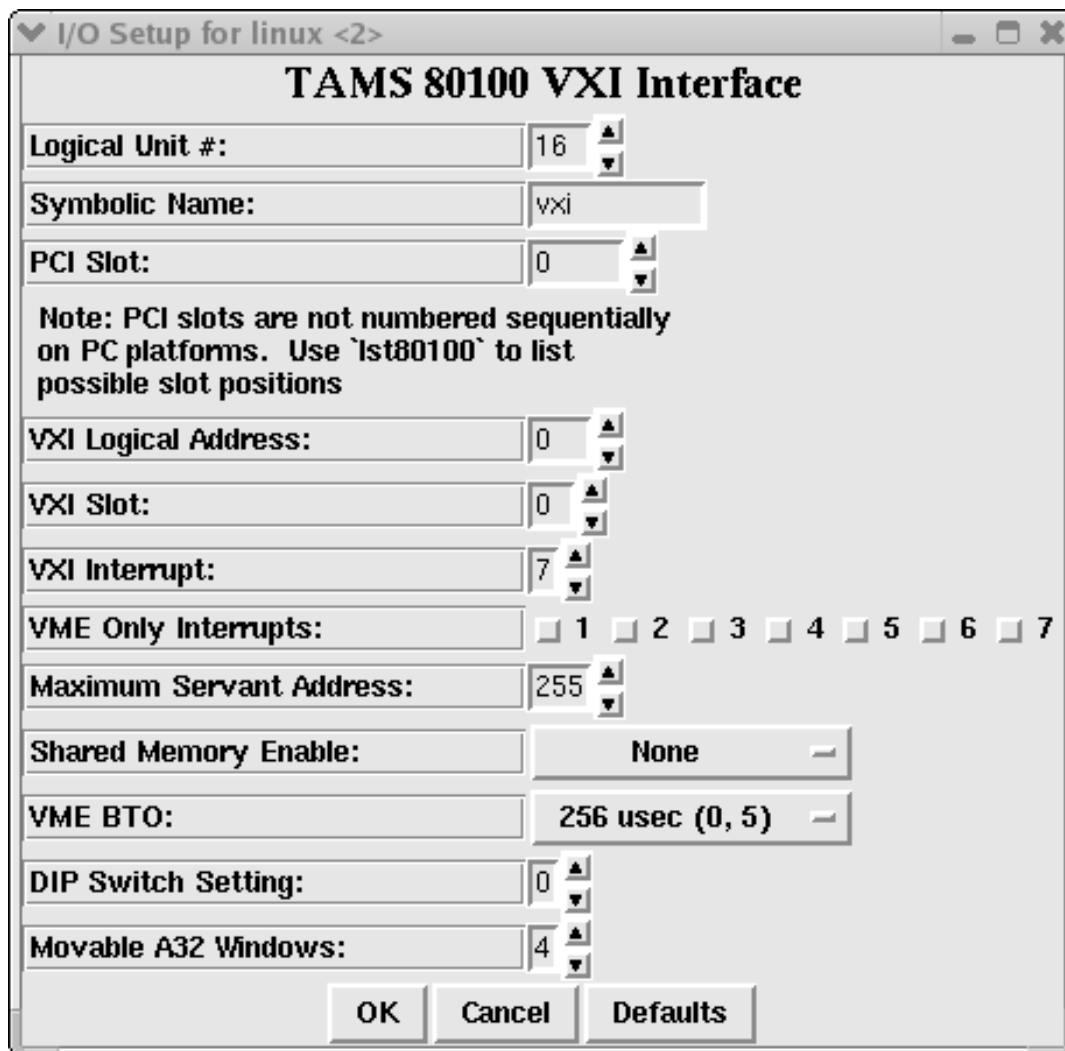


Figure 2. Linux 80100 configuration dialog with default values.

CONFIGURING WITH *HWCONFIG.CF* AND *SICLCONF*

The TAMS 80100 PCI-VXI card can also be configured without the graphical interface. To do so:

ON HP-UX 10.20 AND 11i

1. Edit the *hwconfig.cf* file: `/etc/opt/sicl/hwconfig.cf`.
2. Run `/opt/sicl/bin/siclconf` as root.
3. Edit the *iproc.cf* file as described in the section: *Editing the /etc/opt/sicl/iproc.cf File*.
4. Reboot the workstation manually.

ON LINUX

1. Edit the *hwconfig.cf* file: `/etc/opt/sicl/hwconfig.cf`.
2. Run `/opt/sicl/bin/siclconf` as root.
3. Edit the *iproc.cf* file as described in the section: *Editing the /etc/opt/sicl/iproc.cf File*.
4. Restart the **sicl** service: `/sbin/service sicl stop ; /sbin/service sicl start`.

Entries for the TAMS 80100 in the *hwconfig.cf* file on both HP-UX and Linux have the form:

```
<Logical Unit Number> <Symbolic Name> t80100 <PCI Slot Number> <VXI Logical Address>  
<VXI Slot Number> <80100 Interrupt> <VME Only Interrupts> <Max Servant Addr>  
<Shared Mem Space> <VME BTO> <DIP Switch> <Moveable Windows>
```

While this line is too long to be printed on one line in the manual, it must be all on one line in the *hwconfig.cf* file.

See the section *Configuration Parameters* for a full description of the meaning of each parameter.

EDITING THE `/ETC/OPT/SICL/IPROC.CF` FILE

The installation process of the TAMS 80100 does not automatically edit the `/etc/opt/sicl/iproc.cf` file, which is necessary for the proper execution of the ***ipro*** daemon. The ***ipro*** daemon ensures that the Resource Manager runs when the VXI bus is reset.

ON HP-UX 10.20 AND 11i

As root, edit `/etc/opt/sicl/iproc.cf` to ensure that the following lines are uncommented:

```
boot echo "SICL: Instrument I/O Initialization"
boot siclsetup

boot ivxirm -p -I vxi

sysreset vxi ivxirm -t 5 &

monitor
```

Reboot the machine in order to start the *ipro* daemon.

ON LINUX

As root, edit `/etc/opt/sicl/iproc.cf` to ensure that the following lines are uncommented:

```
boot echo "SICL: Instrument I/O Initialization"

boot ivxirm -p -I vxi

sysreset vxi ivxirm -t 5 &

monitor
```

Restart the *sicl* service (which includes the *ipro* daemon), with

```
/sbin/service sicl stop ; /sbin/service sicl start
```

or run ***iosetup*** or ***siclconf***, which will restart the *sicl* service whenever an 80100 PCI-VXI Controller is configured.

USING THE TAMS 80100B PCI-VXI CARD

PROGRAMMING DIFFERENCES

In general, code written for either the V743 or E1489 MXI controller should work without significant modification. There are some differences that may simplify programming for the user.

- **The entire A24 region is always available.** There is no need to map and unmap regions to move windows around. The entire 16MByte range of A24 can be mapped into a single window or multiple mappings can be made. Mappings can overlap.
- **The entire A16 region is always available.** The entire 64KByte range of A16 can be mapped into a single window or multiple mappings can be made. Mappings can overlap.
- **Up to 6 moveable windows in A32.** The first mapping to the window fixes the size of the window. Windows may **not** overlap.
- **All windows allow 32 bit accesses.** It is not necessary to use the `I_MAP_A16_D32`, `I_MAP_A24_D32` or `I_MAP_A32_D32` flags, although there is no harm in doing so.
- **The 80100 implements DMA.** The `iblockcopy()` and `iblockmovex()` commands use hardware DMA and offer substantial throughput increase for large block transactions. The E1489 did not implement DMA, so users of the E1489 will notice a substantial increase in the speed of these commands.

LINUX SPECIFIC ISSUES

- **BIOS issues.** Some BIOSs will not recognize the PCI bridge chip used on the PCI card in the TAMS 80100 product. It appears that this problem is caused by the BIOS expecting this PCI bridge chip to be used only in a RAID controller, resulting in the BIOS treating the card as a mass storage device. We believe this is an issue only with older BIOSs.

NOTE The BIOS should be set for a “non plug & play OS” in the BIOS flash settings.

Install the PCI card in the Linux workstation and boot the machine. After installing the I/O Libraries, type the following at the command line:

```
/opt/sicl/bin/lst80100
```

You should see a display of the PCI locations (composite of bus and device numbers) where the TAMS 80100 PCI card(s) are detected. If nothing is printed, then the BIOS didn't recognize the card(s) and didn't pass the information to the operating system.

NOTE The listing of PCI cards done by the BIOS during bootup is not a concern. The 80100B PCI card(s) may or may not appear in that list. The first 80100 utility provides the definitive answer on whether the card(s) are recognized. It's possible that a failure to recognize the card(s) may be resolved by upgrading the BIOS or by moving the PCI card(s) to a different slot. Ensure that the required minimum revision software is installed. If the system still does not recognize the card(s), then the 80100B card(s) cannot be used in this computer.

- **Byte swapping issues.** De-referencing a pointer returned from the `imap()` function will return byte swapped values for multi-byte variables on an Intel platform (as compared to a HP-UX platform). This affects the `i*peek()` and `i*poke()` functions since they are currently implemented as macros in `sicl.h`.

There are two ways to deal with this.

- **Preferred.** Use the extended VXI commands (`imapx`, `ipeekx8`, `ipeekx16`, `ipeekx32`, `ipokex8`, `ipokex16`, `ipokex32`). These will work correctly regardless of platform and are the preferred solution since some VXI interfaces only support these commands.
- **Compatibility.** The second solution is to use a compatibility macro to keep the `i*peek()` and `i*poke()` commands working.

```
#define _SICL_SWAP_PEEKPOKE
#include <sicl.h>
```

Notice that `sicl.h` is included after defining the macro.

- **Bus Errors.** In HP-UX, a VXI Bus Error would be propagated to the user process by sending a `SIGBUS` to the user process. If the user wanted to do their own signal handling, they had to install their own signal handler to catch `SIGBUS` and do their own processing.

On Linux and Windows, a VXI Bus Error is signaled using the SICL interrupt handling mechanism. One has to install an interrupt handler with `ionintr()` and then check for the reason `I_INTR_VXI_BUSERR`.

VXI Bus Error notification happens asynchronously with the actual attempt to access the VXI bus. If one needs to associate the actual bus access with a potential Bus Error, then one needs to delay after the bus access to allow a potential Bus Error to propagate.

- **Porting between Little Endian and Big Endian systems.** PA-RISC uses Big Endian byte ordering while the Intel architecture uses Little Endian byte ordering. Although the VXI interface hardware will byte swap multi-byte integers to the correct byte ordering, one should be aware that selecting individual bytes within a multi-byte integer will give different results.

For example, on the VXI bus, the 32 bit integer 0x87654321 is stored as:

```
31                                     0
| 0x21 | 0x43 | 0x65 | 0x87 |
```

On a PA-RISC system, since it is also big endian, accessing the bytes within the 32 bit double word with the following code:

```
unsigned long addr = 0x200000;
unsigned long x;
char * c_ptr;
uint16_t w_ptr;

c_ptr = (char *)&x;
w_ptr = (uint16_t *)&x;
ipeekx32(id ,addr ,0x0, &x);
```

yields

```
x = 0x87654321,

c_ptr[0] = 0x87,
c_ptr[1] = 0x65,
c_ptr[2] = 0x43,
c_ptr[3] = 0x21

w_ptr[0] = 0x8765,
w_ptr[1] = 0x4321
```

On an Intel based system, the VXI interface will give a correctly swapped 32 bit double word, but byte accesses will be different. The same code still yields

```
x = 0x87654321,
```

but, due to differences in host byte order, produces

```
c_ptr[0] = 0x21,
c_ptr[1] = 0x43,
c_ptr[2] = 0x65,
c_ptr[4] = 0x87

w_ptr[0] = 0x4321,
w_ptr[1] = 0x8765
```

For code to be portable between Big and Little Endian systems, one should access the VXI bus with the word width one wants to interpret, rather than with a larger access that needs to be parsed. (i.e., if one wants to do two sequential 16 bit access, one should do individual accesses rather than a 32 bit access that is parsed into two 16 bit halves, since the sub-words will not be swapped correctly.)

The following code gives identical results on both platforms:

```
unsigned short y1, y2;  
ipeekx16(id, addr, 0x0, &y1);  
ipeekx16(id, addr, 0x2, &y2);
```

now $y1 = 0x8765$ and $y2 = 0x4321$ on both platforms, since everything is relative to the VXI bus address instead of a local memory organization.

POWER-CYCLING CONSIDERATIONS

- **Powering down the VXI cage.** When powering down the VXI chassis, the Resource Manager will fail with errors:

```
ERROR: vxi: Internal error occurred
SICL (IPROC): ivxirm reports errors. STAT_LOC: 256
```

This correctly indicates that the TAMS 80100 cannot process VXI transactions while powered down, but does not indicate any functional problems.

- **Booting with VXI cage unpowered.** If the workstation is booted without the VXI chassis connected and powered up, then an error message will be generated during the boot process. An error will be logged, and the **dmesg** command will show an error. These are not significant errors; they simply indicate an inability to communicate with the chassis when the system was booted. When the chassis is later powered up, the **iprocc** daemon (if running) will cause the Resource Manager to run and properly initialize the rack.

If **iprocc** is not running, then the Resource Manager should be run manually by use of the **iclear** and **ivxirm** commands. The **ivxirm** command defaults to 0 delay, so use of a “-t5” parameter is recommended.

```
iclear vxi
ivxirm -t5
```

- **Multi-cage power sequencing.** If multiple racks are daisy-chained (see below) power cycling of the rack nearest the workstation (with the 80100 card) causes the Resource Manager to run and initialize the system. However, if racks are cycled that do not contain the 80100 card, then the Resource Manager is not automatically run. With the **iprocc** daemon running, this is properly done by use of the **iclear** command.

MULTI-CAGE CONFIGURATION

The 80100 PCI-VXI is primarily a single cage controller. To control multiple cages with the 80100, TAMS supports multi-cage linking with National Instruments MXI-VXI-2 cards. In this configuration, the TAMS 80100 is used to link a workstation to the first cage; additional cages are linked via a daisy chain of MXI-VXI-2 cards.

MXI-VXI-1 (Agilent E1482) cards are not supported. They violate the VME bus specification in a manner that makes them incompatible with the TAMS 80100.

TAMS 80100B
PCI-VXI Controller
Printed in USA E 07.27.04

80100-90007

Copyright © Test & Measurement Systems Inc. 2000 - 2004