
Design Guidelines for PC Card and CardBus

A Technical Reference for PC Card and CardBus Controllers and Devices for the Microsoft® Windows® Family of Operating Systems

Addendum to *PC 2001 System Design Guide* Version 1.1 – April 12, 2000

Intel Corporation and Microsoft Corporation

The information contained in this document represents the current view of Intel Corporation and Microsoft Corporation on the issues discussed as of the date of publication. Because Intel and Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Intel and Microsoft, and Intel and Microsoft cannot guarantee the accuracy of any information presented. This document is for informational purposes only. INTEL AND MICROSOFT MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS DOCUMENT.

Intel Corporation and Microsoft Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property rights.

Intel and Microsoft do not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Intel and Microsoft disclaim all expressed and implied warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Intel and Microsoft do not make any warranty of any kind that any item developed based on these specifications, or any portion of a specification, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Intel and Microsoft shall not be liable for any damages arising out of or in connection with the use of these specifications, including liability for lost profit, business interruption, or any other damages whatsoever.

Direct3D, DirectDraw, DirectShow, DirectX, Microsoft, MS-DOS, Win32, Windows, Windows NT, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Indeo, Intel, and Pentium are registered trademarks of Intel Corporation.

Other product and company names herein may be the trademarks of their respective owners.

© 1998-2000 Intel Corporation and Microsoft Corporation. All rights reserved.

Revision History

Revision	Publication Date	Comments
1.00	March 20, 2000	Release. Extracted from PC 2001 System Design Guide, rev. 0.7.
1.1	April 12, 2000	Clarifications: PCCard-2: Reinstated PCCard-8: Required for PC 2001 PCCard-19: Removed "or later" from <i>PC Card Standard, Release 7</i> cite PCCard-22: Terminology change: LVE to VPE Added trademark for DirectDraw

Introduction

This design guide — which is provided as an addendum to *PC 2001 System Design Guide* — presents requirements for PC Card controllers and devices designed to work with the Microsoft® Windows® family of operating systems. This includes requirements for 16-bit PC Card, previously referred to as Personal Computer Memory Card International Association (PCMCIA) cards, CardBus cards, and PC Card socket controllers.

This material was originally published in *PC 99 System Design Guide*, co-authored by Intel and Microsoft. Because the support in the Windows family of operating systems has not changed, and is not expected to change in the future, this addendum has been created as a stand-alone reference.

Please note that the requirements described in this addendum are incorporated as part of the PC 2001 requirements by reference from within *PC 2001 System Design Guide*.

Note: PC Card requirement changes from *PC 99 System Design Guide* are indicated as Notes in this guide. These changes include the following:

- Clarification of CardBus power management requirements, with definition of requirements for wake-from-D3cold and Vaux (requirement 19)
- Update of specification references for PC Card Standards
- Clarification of ZV requirement: changed to “if implemented” (requirement 2)

Microsoft Windows Millennium Edition (Windows Me), Windows 98, and Windows 2000 Professional support 16-bit PC Card I/O cards and CardBus I/O cards. Memory 16-bit PC Card cards are supported only as legacy devices. For any PC Card device to work effectively with Windows 98/Me or Windows 2000, the manufacturer must implement a minimum set of tuples documented in the PC Card Standard. Windows uses these tuples to identify and configure any 16-bit PC Card card, and it might also use these tuples for CardBus cards.

PC Card Requirements

This set of guidelines summarizes requirements for PC Card implementation. These requirements apply to two distinct types of PC Cards: 16-bit PC Cards, and CardBus cards (which are 32-bit cards).

For each of these two types of PC Cards, requirements are defined in four different implementation areas:

- Socket controller (“bridge”) requirements
- Host-system support requirements (such as system board and firmware)
- Software requirements

- Card requirements

Note that power management requirements may be included in each of these requirements categories, in keeping with the fact that power management is a system-level capability. Also note that both 16-bit PC Cards and CardBus cards can be multifunction cards (that is, cards that include multiple devices, all of which share that card's single PC Card interface).

Each device in a multifunction PC Card—such as a CardBus card—must separately meet the power management device class specifications for its device class and be independently power managed. This means that it is not necessary for both device A and device B on the same add-on card to be idle before the system can change the power-state of one or both of these devices.

A CardBus socket can accept any of the previously mentioned PC Card types. When a CardBus socket controller interfaces with a 16-bit PC Card, the controller works differently than when interfacing a CardBus card. The list of CardBus requirements includes accommodation of 16-bit PC Cards by a CardBus socket controller

PC Card Basic Requirements

This section summarizes the basic requirements for PC Card.

PCCard-1. All devices comply with the PC Card standards

Required — PC 99:12.1

Designs for PC Card socket controllers and cards must all be based on *PC Card Standard Release 7* or later.

All PC Card devices must comply with these standards.

For information about implementing R2 version cards to use only 3.3 volts, see the white paper titled “Card Voltage Requirements and the Windows Operating System” at <http://www.microsoft.com/hwdev/cardbus/pccardvlt.htm>.

Note: In *PC 99 System Design Guide*, this requirement referred to earlier versions of the standard. This guide cites the current specification. The implementation requirements remain the same.

PCCard-2. System and ZV-compatible 16-bit PC Cards comply with ZV standard definitions

Required - PC 99:12.2

The PC Card standards define the requirements for Zoomed Video (ZV) cards and system support.

Note: In *PC 99 System Design Guide*, Zoomed Video support was required. Zoomed Video is not required for PC 2001, but if it is implemented, it must support PC Card standards.

PC Card Socket Controller Requirements

This section summarizes requirements and standards for socket controllers.

PCCard-3. Controller supports industry-standard ExCA register set

Required — PC 99:12.3

The built-in software supporting 16-bit PC Card cards in Windows includes drivers for the industry-standard Exchangeable Card Architecture-compatible (ExCA-compatible) socket controllers. To be compatible with these drivers, socket-controller implementations must support the industry-standard ExCA base register set.

Notice that some controllers do not fully implement the register set and therefore are incompatible. Also, some controllers implement extended registers or enhancements. The built-in Windows drivers do not exploit these features, even though the controller might be compatible.

PCCard-4. System maintains mapping of IRQ Routing Register bits to system interrupt vectors

Required — PC 99:12.4

The system design must maintain the mapping of the PC Card controller's IRQ Routing Register bits to system interrupt vectors. This means that when an interrupt is programmed in the controller to occur on the IRQ_x pin, the system's IRQ routing causes the interrupt controller to generate the interrupt vector for IRQ_x and no other IRQ.

PCCard-5. IRQ connections can be determined by using the 0805 register

Required — PC 99:12.5

Windows uses the 0805 register on CardBus controllers to determine which ISA IRQs are connected to the controller. This register must engage (drive low when the IRQ is asserted) the corresponding ISA IRQ when programmed with a value. It must deselect the IRQ (float high) when programmed at zero (0). This behavior must be achieved without requiring the operating system to program any non-standard registers.

PCCard-6. CardBus controllers support both ISA and PCI interrupts**Required — PC 99:12.6**

PC Card software dynamically configures the bridge to use ISA interrupts for 16-bit PC Card cards and to use Peripheral Component Interface (PCI) interrupts for CardBus cards. As defined in earlier requirements, CardBus controllers must maintain mapping of IRQ routing. Also, notice that systems implementing CardBus controllers must fully support PCI 2.2 as well as additional PCI requirements for IRQ routing as defined in *PC 99 System Design Guide* and *PC 2001 System Design Guide*.

To ensure that the Windows operating system can correctly assign ISA IRQs to 16-bit PC Cards, CardBus controllers that have parallel ISA IRQ mode must have all ISA IRQs pins, except IRQ 0 (timer), 1 (keyboard), 6 (floppy), 8 (CMOS), 13 (math coprocessor). It is recommended that system vendors using parallel ISA IRQ mode always connect ISA IRQs 3, 4, 5, 7, 9, 10, 11, 12, 14, 15 and not cross wired them.

Note: Vendors using serialized IRQ mode only need to connect the serial IRQ pin, and the ISA IRQ information will be sent to the PCI chip set serially; the ISA IRQ information can specify any of IRQ 0–15.

PCCard-7. System supports industry-standard definition for CardBus bridges**Required — PC 99:12.7**

Systems must support the definition in *PC Card Standard Release 7* (or later) *PC Card Host System Specification (Volume 11)*, *PCI-to-CardBus Bridge Register Description (Section 4)* for CardBus controllers (PCI-to-CardBus bridges). This definition includes a common PCI Configuration Space header assigned the Header Type field value of 82h.

Windows supports this specification. Any controller features that are not part of this specification will not be used in standard drivers. The BIOS is responsible for any hardware initialization or setup required to make the controller comply with this specification or other requirements in this document.

Because CardBus host controllers are PCI bus bridges, they will be supported (enumerated and configured) by the PCI software in Windows in the same manner as other PCI bus bridges.

Note: In *PC 99 System Design Guide*, this requirement referred to *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification). This guide cites the current specification as it was incorporated into the PC Card standard. The implementation requirement remains the same.

PCCard-8. BIOS initializes CardBus controller in 82365-compatible mode and supports backward compatibility**Recommended — PC 99:12.8**

Note: In *PC 99 System Design Guide*, this item was recommended. It is required for PC 2001.

When Intel 82365-compatible modes are implemented, CardBus controllers are enumerated and configured in the same way as other PCI bus bridges. The PCI bus bridge support in Windows 98 is based on requirements for PCI interrupt routing and bridge-window configuration. Therefore, full compliance with the latest PCI specifications is required for CardBus support.

There are steps the BIOS can take to achieve backward compatibility with Windows. Specifically, the BIOS can initialize the CardBus controller in Intel 82365-compatible mode and report it as device “PNP0E03, Intel 82365-compatible CardBus controller.” The requirements are as follows for BIOS POST time (CardBus controller ConfigSpace initialization):

- Command register (offset 0x04) set to 0x07 (IOspaceEnable, MemSpaceEnable, BusMasterEnable).
- RegisterBaseAddress (offset 0x10) set to 0. If support for other environments is needed, such as Windows 3.1 or MS-DOS®, some other value can be set.
- All memory and I/O windows (offset 0x1c–0x38) set to 0.
- Interrupt Line register (offset 0x3c) set to 0xff (no IRQ assigned). If support for other environments is needed, such as Windows 3.1 or MS-DOS, an assigned IRQ line can be set. Notice, however, that this register must be set to 0xff at the time that the device is disabled by the operating system, and then set into CardBus mode. More information about BIOS enumeration is presented later in this requirement.
- Other controller-specific initialization as required to put the controller in legacy mode.

This puts the CardBus controller into legacy mode where the Windows Socket Services driver can access it as an Intel PC Card I/O card-compatible (PCIC-compatible) controller at an I/O address, for example, 0x3e0.

Notice that the BIOS must be at least PCI 2.2-compliant and must support the \$PIR Interrupt Routing Table. The \$PIR table must return the necessary PCI IRQ routing information, including the routing information for the CardBus controller. In general, if the CardBus controller is on the system board, there must be a slot routing entry for it in the table. If the CardBus controller is a PCI add-on card, there must be routing information entries for each PCI slot in the system.

During Plug and Play BIOS enumeration, the BIOS should report the CardBus controller as *pnp0e03 with a compatible ID of *pnp0e00 and the I/O resource of two ports, for example, 0x3e0–0x3e1.

For more information, see the white paper on CardBus host controllers and Windows compatibility at <http://www.microsoft.com/hwdev/cardbus/>.

PCCard-9. CardBus controllers do not share writable PCI Configuration Space bits

Required — PC 99:12.9

CardBus controllers are multifunction PCI devices, and Windows treats each function as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits, such as the Command register.

Notice that the 16-bit PC Card interface legacy-mode Base Address Register (BAR; offset 44h in the Type 2 PCI header) is the only exception to this requirement. This BAR must be shared between the two functions in order to be compatible with the ExCA programming model.

PCCard-10. Each 16-bit PC Card memory window in CardBus controller has its own page register

Required — PC 99:12.10

For complete flexibility and support of typical configurations, CardBus controllers must support the independent location of R2 memory windows anywhere in the full system address space as recommended in the *PC Card Standard Release 7* (or later), *PC Card Host System Specification*, *PCI-to-CardBus Bridge Register specification*.

Controllers that share a single page register among all 16-bit PC Card memory windows require that all 16-bit PC Card memory windows must be located within the same 16-MB block. This is often not possible with typical (16 MB) DRAM and bridge (positive-decode) configurations. The result is disabled cards.

Note: In *PC 99 System Design Guide*, this requirement refers to the Yenta specification. This guide cites the current specification as it was incorporated into the PC Card standard. The implementation requirement remains the same.

Plug and Play Design for 16-bit PC Card Cards

This section summarizes the Plug and Play requirements for 16-bit PC Card cards.

The Windows operating system determines what type of card is plugged into the PC Card socket by examining the tuples on the card. For Plug and Play functionality, 16-bit PC Card I/O cards must support a set of required information and configuration tuples. The PCMCIA bus enumerator uses these tuples to

identify the card, load the correct device driver, and indicate all possible configurations to the Plug and Play configuration manager. The operating system then dynamically assigns a valid configuration based on this information.

PCCard-11. Card supports required I/O card tuples

Required — PC 99:12.11

The following items must be implemented for any 16-bit PC Card I/O card that connects to a system:

- The 16-bit PC Card card must contain:
 - The device information tuple (CISTPL_DEVICE, 01h for cards capable of 5V operation or CISTPL_DEVICE_0C, 1Ch for cards capable of 3.3V operation).
 - The Level 1 (L1) version/product information tuple (CISTPL_VERS_1, 15h).
 - The configuration tuple (CISTPL_CONFIG, 1Ah).
 - The configuration table entry tuple (CISTPL_CFTABLE_ENTRY, 1Bh).
- A 16-bit PC Card card with more than 64 MB Common Memory must contain the extended device information tuple (CISTPL_EXTDEVICE, 09h).
- The L1 version/product information tuple must contain the product name and manufacturer name in the product information string (TPLL1V1_INFO, byte 4).
- The product name and manufacturer name in the L1 version/product information tuple must be composed only of ASCII characters greater than ASCII 20h and less than ASCII 7Fh.

Windows uses the information contained in the required and recommended tuples to create a unique device ID for the card and to assimilate configuration information for the device. Windows uses the device configuration tuples to determine the general characteristics of the card.

Required I/O Card Tuples

Tuple ID	Tuple code	Description and comments
01h	CISTPL_DEVICE	Device information (common memory). For non-memory cards, this tuple must be present, but the device type will be NULL.
15h	CISTPL_VERS_1	L1 version/product information strings: Product information, Product name, Product number, Other manufacturer information
1Ah	CISTPL_CONF	Configuration. Indicates the location of configuration registers and registers present.
1Bh	CISTPL_CE	Configuration table entry. Appropriate configuration requirements for I/O space, interrupts, memory, and so on should be specified.

Tuple ID	Tuple code	Description and comments
20h	CISTPL_MANFID	Manufacturer ID. Card manufacturer ID code. Defines manufacturer for this card.
21h	CISTPL_FUNCID	Function ID. Provides function information about the card. Also includes system initialization information.

The device information tuple provides information about the memory devices used in the card's common memory space. The device type, size, and speed are used to configure the socket for efficient access to the card. This tuple must be present on 16-bit PC Card I/O cards, but the device type must be NULL.

The L1 version/product information tuple contains human-readable information about the product and its manufacturer. This information is intended to be displayed to the user where necessary. Windows uses the information contained in the product information string and product name string to construct the device ID for that card. It also scans through the tuple, starting at the very beginning and continuing to the end of the product name string.

The information gathered from the L1 version/product information tuple is used to construct the unique device ID. Because the optional third and fourth strings in the tuple are not used in the unique ID, devices that require unique numbers on each card can use these strings to store that information.

The configuration tuple tells the software where to locate the configuration registers that program the card's configuration, as well as which registers are present on the card.

Each configuration table entry tuple completely describes one valid configuration in which the card can operate. Each entry describes power, timing, I/O space, interrupt, and memory space requirements for the given configuration. Configuration software selects one of these configurations for the card based on the resources currently available in the system.

The manufacturer ID tuple (CISTPL_MANFID, 20h) and the function ID tuple (CISTPL_FUNCID, 21h) add extra flexibility to a PC Card that connects to the PC:

- The manufacturer ID tuple provides unique information about the card manufacturer. This code is registered with PCMCIA. Windows uses the manufacturer ID tuple as one source for creating a 16-bit CRC used in the construction of the device ID.
- The function ID tuple provides information about the class of device or what function the card provides, for example, memory, modem, disk, and so on. This information helps the software perform necessary installation tasks and locate compatible drivers. Although it is not required to make this determination, Windows uses the function ID tuple internally to determine what type of device is on the PC Card.

PCCard-12. Configuration table entry tuples listed in priority order**Required — PC 99:12.12**

Configuration table entry tuples are placed in the preferred order for configuring the device. Windows processes the tuples in the order they are placed in the Card Information Structure (CIS). From these tuples, Windows creates a logical configuration in this order and prioritizes them in the same order. Notice that for multiple voltage cards, the voltage policy is to prioritize 3.3-volt configurations, if they are supported by the system, over 5-volt configurations, regardless of the order of the configuration table entry tuples (CISTPL_CFTABLE_ENTRY).

PCCard-13. Card specifies maximum configuration options**Required — PC 99:12.13**

Many older PC Cards specified fixed configurations in order to address compatibility with existing software. However, this is not the intended use for tuples; the configuration software should be responsible for compatibility. The tuples should be used only to describe its maximum configurability, ruling out configurations not supported by the hardware.

If fixed configurations must be provided for an operating system other than Windows, there must be one or more entries that specify the maximum configurability that the hardware can handle. An example of “maximum configurability” is to specify “any IRQ” rather than only IRQ 3 or IRQ 4.

Plug and Play Design for CardBus

This section summarizes the Plug and Play requirements for CardBus cards. CardBus was designed as a combination of the 16-bit PC Card and PCI. The goal is to gain the benefits of PCI in a PC Card format. Consistent with this goal, Windows support for CardBus places specific requirements on CardBus cards.

PCCard-14. Configuration Space meets Common Silicon Guidelines**Required — PC 99:12.14**

The Common Silicon Guidelines are defined in Section 2.1 of the *PC Card Standard Guidelines, Volume 10*.

Note: *PC 99 System Design Guide* cited Section 2.6. This guide corrects the citation and provides the following additional notes:

- The standard for CardBus defines a PCI “Type-2” configuration space that is defined in Section 4.5 of Volume 11 (*PC Card Host System Specification*) of the *PC Card Standard Release 7*. The Type-2 CardBus-bridge PCI header structure was defined to be as similar to the Type-1 (PCI-to-PCI bridge) header as possible. Type-2 and Type-1 PCI headers differ only in that the

Type-2 header allows 4-byte resolution in I/O Base and Limit registers, while the Type-1 header supports a coarser 4K resolution for these registers.

- CardBus cards include normal Type-0 PCI headers, with certain provisions. The quadword register located at 0x28 is used as a pointer to the CardBus Card Information Structure (CIS). CardBus cards must also implement certain Command and Status Register fields that are optional for PCI devices. CardBus cards must also provide a Memory BAR for every I/O BAR provided (so that I/O windows can be memory-mapped).
- Section 2.1.3.4 of Volume 10 (*Guidelines*) of the *PC Card Standard Release 7* details the common silicon guidelines to which CardBus cards must adhere.

To maintain compatibility with existing PCI system software and drivers, Windows will support only CardBus cards whose Configuration Space is designed to meet the Common Silicon Guidelines. This is a requirement because CardBus configuration is performed by the PCI software, which can deal with all aspects of PCI topology configuration, including bridging. Without the allocated fields, the cards cannot be fully treated as PCI devices and cannot be supported under Windows.

The required allocated fields are listed in the following table.

Required Allocated Fields

Field	Description and comments
Vendor ID	This read-only field contains a unique ID (in PCI space) for the card manufacturer. The PCI SIG allocates unique IDs.
Device ID Revision ID	These read-only fields are vendor-assigned values that uniquely identify the device (among all vendors of PCI or CardBus products).
Class Code	This read-only field is defined in PCI 2.2. It describes what type of device the card is.
Max_Lat Min_Gnt	These read-only fields specify the desired settings for Latency Timer values according to PCI 2.2. A value of 0 indicates the device has no major requirements for the settings of Latency Timers.
Interrupt Line	This register must be read-write and must not be connected to anything, just as on PCI cards. This register is used to store the current IRQ routing for the device.

PCCard-15. RESERVED fields comply with PCI 2.2

Required — PC 99:12.15

The CardBus specification also lists two RESERVED fields (offset 2C in the Configuration Space), which have since been defined in PCI 2.2. These fields are also required on CardBus cards for Windows compatibility.

Required RESERVED Fields

Field	Description
Subsystem ID	If different from Device ID
Subsystem Vendor ID	If different from Vendor ID

PCCard-16. CardBus card implements required and recommended tuples**Required — PC 99:12.16**

For CardBus, Windows also requires the same set of card tuples recommended in the PC Card guidelines, as summarized in the following table.

Required CardBus Tuples

Tuple ID	Tuple code	Comments
04h	CISTPL_CONFIG_CB	—
05h	CISTPL_CFTABLE_ENTRY_CB	—
07h	CISTPL_BAR	—
13h	CISTPL_LINKTARGET	Required as first tuple in PC Card standard.
15h	CISTPL_VERS_1	—
20h	CISTPL_MANFID	—
FFh	CISTPL_END	Required as end-of-chain tuple in PC Card standard.
21h	CISTPL_FUNCID	Recommended in PC Card standard; required for Windows operating system compatibility.

Requirements for PC Card

This section summarizes additional requirements for PC Card.

Power Management for PC Card

This section summarizes the specific power management requirements for PC Card. Power management requirements for specific device classes are defined in the related chapters in *PC 99 System Design Guide* and *PC 2001 System Design Guide*.

PCCard-17. Socket controller complies with device class power management reference specification**Required — PC 99:12.17**

This applies for both 16-bit PC Card-only controllers and CardBus controllers.

The *PC Card Controller Device Class Power Management Reference Specification, Version 1.0* or later, provides class-specific definitions of the OnNow device power states (D0–D3) for these devices. The specification also covers device functionality expected in each power state and the possible wake-up event definitions for the class, for example, whether card insertion should wake the system.

PCCard-18. 16-bit PC Card cards implement power-related events using ReqAttn bit and #STSCHG mechanism**Required — PC 99:12.18**

Any 16-bit PC Card card that is capable of signaling a wake-up event to the system, as defined in the device class power management reference specification for its class, must implement the ReqAttn bit and its associated enable bit in the Extended Status register, and must signal on the #STSCHG line.

PCCard-19. CardBus controllers and cards implement PCI and CardBus power management specifications**Required — PC 99:12.19**

PCI-to-CardBus bridges and CardBus cards must comply with the requirements defined in *Section 3 (PCI Bus Power Management Interface Specification for PCI-to-CardBus Bridges)* of the *PC Card Standard, Release 7*. This specification describes the CardBus power-management interface hardware, as well as proper software use of these hardware mechanisms.

The CardBus card must use the CSTSCHG pin to signal wake-up events. This is because there is no PME# pin on the CardBus interface, and the CardBus card must use PME_EN in the card's Configuration Space to enable wake-up events. Specifically, setting the PME_EN bit in the card's Configuration Space must provide the same behavior as setting both the GWAK and WKUP bits in the card's Function Event Mask register.

If wake-from-D3cold is implemented in a platform, the following are required:

- Associated CardBus controller must support PME# assertion from D3cold.
- Associated socket must supply sufficient Vaux power to support the card in its D3cold state.

This requirement must be independently met by each enabled D3cold-wake-capable CardBus socket in the system, as defined in the host system chapter of PC Card Standard, version 7.

Power management requirements for 16-bit PC Card cards are defined earlier in requirement 18.

Note: This guide defines these new power management requirements over what was required in *PC 99 System Design Guide*:

- *Section 3 (PCI Bus Power Management Interface Specification for PCI-to-CardBus Bridges)* of the *PC Card Standard, Release 7* as a new specification for CardBus cards. (PC 99 requirements called only for PCI-PM 1.0.)
- Wake-from-D3cold support for controller and socket.

Device Drivers and Installation for PC Card

This section summarizes requirements for PC Card device drivers.

PCCard-20. No user intervention required for correctly installing devices

Required — PC 99:12.20

The user must not be required to perform any device-installation action other than to insert disks that contain drivers and other files.

PCCard-21. Device is immediately functional without restarting the system

Required — PC 99:12.21

The user must be able to begin using the device without having to restart the system. Device use begins either after installation is complete or whenever the device is inserted in the system.

PCCard-22. ZV-compatible PC Card driver uses DirectDraw VPE

Required — PC 99:12.22

ZV-compatible PC Card drivers must use software interfaces based on 32-bit Microsoft DirectDraw® Video Port Extensions (VPE) in order to configure the graphics controller to receive video input using the ZV port. This includes programming the graphics controller to configure the format of the video data, its location on screen, and so on. VPE is part of Microsoft DirectX® APIs.

ZV card device drivers must handle dynamic graphics state changes, such as resolution changes, color depth changes, and switching to and from full-screen MS-DOS®-based applications.

Note: Since *PC 99 System Design Guide*, the extensions to DirectDraw APIs identified as Live Video Extensions (LVE) have been renamed Video Port Extensions (VPE). The implementation requirement remains the same.

PCCard-23. 16-bit PC Card card driver supports sharing of level-mode interrupts**Required — PC 99:12.23**

CardBus systems support both 16-bit PC Card cards and CardBus cards. In this environment, interrupt sharing becomes an issue because CardBus controllers can use PCI interrupts, which are level-sensitive and sharable. To help alleviate interrupt limitations in CardBus systems, Windows operating systems can take advantage of PCI interrupt-sharing capabilities.

In cases where no ISA IRQs are available to a 16-bit PC Card card in a CardBus controller, the operating system will assign a PCI interrupt to the card. All 16-bit PC Card card drivers must “hook” the interrupt, whether it is sharable or not, before its hardware generates any interrupts.

PC Card References

The following represents some of the references, services, and tools available to help build hardware that is optimized to work with Windows operating systems.

Microsoft Windows 98 DDK and Windows 2000 DDK

<http://www.microsoft.com/ddk/> or MSDN Professional membership

PC 2001 System Design Guide and *PC 99 System Design Guide*

<http://www.pcdesguide.org>

PC Card Controller Device Class Power Management Reference Specification, Version 1.0 or later

<http://www.microsoft.com/HWDev/specs/PMref/PMcard.htm>

PCI Bus Power Management Interface Specification for PCI to CardBus Bridge, Revision 1.0

PCI Local Bus Specification, Revision 2.2 (PCI 2.2)

<http://www.pcisig.com>

PC Card Standard Release 7

PCMCIA

<http://www.pc-card.com/bookstore.htm>

CardBus host controllers and Windows compatibility white papers

<http://www.microsoft.com/hwdev/cardbus/>

Checklist for PC Card

If a recommended feature is implemented, it must meet the requirements for that feature as defined in this document.

PCCard-1. All devices comply with the PC Card standards

Required — PC 99:12.1

PCCard-2. System and ZV-compatible 16-bit PC Cards comply with ZV standard definitions

Required - PC 99:12.2

PCCard-3. Controller supports industry-standard ExCA register set

Required — PC 99:12.3

PCCard-4. System maintains mapping of IRQ Routing Register bits to system interrupt vectors

Required — PC 99:12.4

PCCard-5. IRQ connections can be determined by using the 0805 register

Required — PC 99:12.5

PCCard-6. CardBus controllers support both ISA and PCI interrupts

Required — PC 99:12.6

PCCard-7. System supports industry-standard definition for CardBus bridges

Required — PC 99:12.7

PCCard-8. BIOS initializes CardBus controller in 82365-compatible mode and supports backward compatibility

Recommended — PC 99:12.8

PCCard-9. CardBus controllers do not share writable PCI Configuration Space bits

Required — PC 99:12.9

PCCard-10. Each 16-bit PC Card memory window in CardBus controller has its own page register

Required — PC 99:12.10

PCCard-11. Card supports required I/O card tuples

Required — PC 99:12.11

PCCard-12. Configuration table entry tuples listed in priority order

Required — PC 99:12.12

PCCard-13. Card specifies maximum configuration options

Required — PC 99:12.13

PCCard-14. Configuration Space meets Common Silicon Guidelines

Required — PC 99:12.14

PCCard-15. RESERVED fields comply with PCI 2.2

Required — PC 99:12.15

PCCard-16. CardBus card implements required and recommended tuples

Required — PC 99:12.16

PCCard-17. Socket controller complies with device class power management reference specification

Required — PC 99:12.17

PCCard-18. 16-bit PC Card cards implement power-related events using ReqAttn bit and #STSCHG mechanism

Required — PC 99:12.18

PCCard-19. CardBus controllers and cards implement PCI and CardBus power management specifications

Required — PC 99:12.19

PCCard-20. No user intervention required for correctly installing devices

Required — PC 99:12.20

PCCard-21. Device is immediately functional without restarting the system

Required — PC 99:12.21

PCCard-22. ZV-compatible PC Card driver uses DirectDraw VPE

Required — PC 99:12.22

PCCard-23. 16-bit PC Card card driver supports sharing of level-mode interrupts

Required — PC 99:12.23