PC CARD STANDARD

Volume 10

Guidelines

PCMCIA
JEIDA

©1999, PCMCIA/JEIDA
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording or otherwise, without prior written permission of PCMCIA and JEIDA. Printed in the United States of America.

PCMCIA (Personal Computer Memory Card International Association)
2635 North First Street, Suite 209
San Jose, CA 95134 USA
+1-408-433-2273
+1-408-433-9558 (Fax)

JEIDA (Japan Electronic Industry Development Association)
Kikai Shinko Kaikan, 3-5-8, Shibakoen
Minato-ku, Tokyo 105, JAPAN
+81-3-3433-1923
+81-3-3433-6350 (Fax)

The PC Card logo and PC Card are trademarks of PCMCIA, registered in the United States. The PC Card logo and design are trademarks of JEIDA, registered in Japan.

PCMCIA HAS BEEN NOTIFIED BY CERTAIN THIRD PARTIES THAT THE IMPLEMENTATION OF THE STANDARD WILL REQUIRE A LICENSE FROM THOSE THIRD PARTIES TO AVOID INFRINGEMENT OF THEIR RIGHTS. PCMCIA HAS OBTAINED FROM SOME, BUT NOT ALL, OF THOSE PARTIES A GRANT OF IMMUNITY THAT PCMCIA WILL EXTEND TO YOU, CONTINGENT UPON YOUR ENTERING INTO AND DELIVERING TO PCMCIA THE RECIPROCAL GRANT OF IMMUNITY AGREEMENT CONTAINED ELSEWHERE IN THIS STANDARD.

IMPORTANT:
In order to receive the Grant of Immunity, the owner of this Standard must sign and return the enclosed Registration Card to:
PCMCIA
2635 North First Street, Suite 209
San Jose, CA 95134 USA

NEITHER PCMCIA NOR JEIDA MAKES ANY WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE STANDARD, INCLUDING AS TO NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THIS STANDARD IS PROVIDED TO YOU "AS IS."

Document No. 0299-10-2000

First Printing, February 1999

CONTENTS

© 1999 PCMCIA/JEIDA

©1999PCMCIA/JEIDA

©1999PCMCIA/JEIDA

© 1999 PCMCIA/JEIDA

FIGURES

© 1999 PCMCIA/JEIDA

TABLES

© 1999 PCMCIA/JEIDA

 ©1999PCMCIA/JEIDA

1. INTRODUCTION

1.1 Purpose

This document is designed to provide implementation examples and further explanations of the ***PC Card Standard*** in order to:

- enhance the interoperability of PC Card components, including card hardware and software, system hardware and software, and applications.
- facilitate the development of PC Card hardware and software by increasing the understanding of the standard by PC Card implementation community.

These guidelines are not requirements made by the PCMCIA/JEIDA standards organizations. Rather, they are implementation examples, suggestions and hints.

1.2 Scope

Guidelines may address any of the following areas:

1. difficult portions of the standard which could use further explanation
2. implementation hints and considerations which a PC Card designer wishes to make public
3. examples which other implementers could use as starting points for their own designs
4. PC Card product areas for which no PC Card Standards exist, e.g., applications
5. interaction of PC Card designs with other areas, e.g., other buses

Guidelines may cover any PC Card hardware or software, whether system or card-based. A guideline may restrict itself to a given architecture, operating system, or other implementation constraint in order to be more effective.

In summary, this guidelines section is meant to be a useful adjutant to the ***PC Card Standard*** and not to be an additional source of such standards.

© 1999 PCMCIA/JEIDA

1.3 Related Documents

The following documents which comprise the ***PC Card Standard***:

PC Card Standard~~,~~ Release 7.0 (February 1999)~~March 1997~~, PCMCIA/JEIDA
Volume 1. ***Overview and Glossary***
Volume 2. ***Electrical Specification***
Volume 3. ***Physical Specification***
Volume 4. ***Metaformat Specification***
Volume 5. ***Card Services Specification***
Volume 6. ***Socket Services Specification***
Volume 7. ***Media Storage Formats Specification***
Volume 8. ***PC Card ATA Specification***
Volume 9. ***XIP Specification***
Volume 10. ***Guidelines***
~~Volume 11. *PCMCIA Specific Extensions*~~
~~Volume 12. *JEIDA Specific Extensions*~~
Volume 11~~3~~: ***PC Card Host System Specification***

MIL-STD-202F, Military Standard, ***Test Methods for Electrical Connectors***, U.S. Department of Defense.

MIL-STD-1344A, Military Standard, ***Test Methods for Electrical Connectors***, U.S. Department of Defense.

ANSI/UL 94-1979, November 16, 1979, ***Standard for Tests for Flammability of Plastic Materials for Parts in Devices and Appliances***

ANSI/EIA-364-65-1992, March 6, 1992, EIA Standard TP-65, ***Mixed Flowing Gas***, Electronic Industries Association

CB14, June 1993, EIA Engineering Bulletin, ***Contact Lubrication***, Electronic Industries Association

EIA-364B, August 1990, ***Electrical/Socket Test procedures including Environmental Classifications***, Electronic Industries Association

IEC Standard No. 950, Second Edition 1991, ***Safety of Information Technology Equipment, including Electrical Business Equipment***, International Electrotechnical Commission.

1.4 Guidelines Format

~~This section is a compendium of individual guidelines.~~ Each guideline has subsections as follows:

Summary	indicates the objective of the guideline and the result of following it
Background	explains purpose of the guideline. In some cases it also gives information on current implementations
Guideline	gives the actual guideline

©1999PCMCIA/JEIDA

2. ELECTRICAL GUIDELINES

2.1 CardBus/PCI Common Silicon Requirements

2.1.1 Summary

One of the goals of the CardBus specification is to allow a component to be designed so that it can function either in the PC Card environment when mounted on a CardBus PC Card or in the PCI environment when attached directly to a PCI bus. While the CardBus specification makes it possible to do this, this guideline provides specific details of how the design must be done in order to achieve this goal.

2.1.2 Background

The PCI specification, Rev 2.0, served as the starting point in developing the CardBus interface. Any differences between the two standards were driven primarily by differences in the electrical environment (the PC Card's 68-pin connector vs. PCI's 120 pin), the point-to-point interconnect mandated by 16-bit PC Card support (vs. PCI's bused environment), and the dynamic insertion/removal characteristics inherent to PC Card but not PCI. This guideline describes the tradeoffs made to overcome these differences and documents how common silicon should be designed and used in the CardBus environment.

The information in the guideline is structured in similar fashion to the CardBus specification to make it easier to identify where CardBus differs from PCI. As a result, this guideline covers the following areas:

1. pin definition differences
2. functional differences
3. electrical differences
4. configuration space differences
5. software driver differences

This guideline should not be used as a substitute for either the CardBus or PCI specifications by design engineers. Its only purpose is to highlight differences, explain why the difference exists, and show what CardBus implementations must do to overcome them. The detail contained is not sufficient to enable a design to be undertaken.

2.1.3 Guideline

2.1.3.1 Pin Definition Differences

The pinout differences between CardBus and PCI are:

1. CardBus does not have an **IDSEL** signal.

   PCI uses **IDSEL** as a chip select for configuration read and configuration write cycles. Since CardBus PC Cards always operate in a point-to-point environment, they are always the

© 1999 PCMCIA/JEIDA

intended target of configuration cycles on the interface. Therefore, a CardBus PC Card will always claim a configuration cycle.

Common silicon components must implement **IDSEL** in order to meet PCI requirements. This signal should be strapped high (always asserted) when mounted on a CardBus PC Card. This will cause the component to be selected whenever a configuration cycle is executed on the CardBus interface.

2. CardBus does not have the **SBO#** or **SDONE** signals.

   PCI uses these optional pins to signal address snooping results to the target of a transaction. The target continues the access when CLEAN is signaled (**SBO#** deasserted and **SDONE** asserted) or terminates with retry when HITM is signaled.

   The CardBus adapter can take appropriate action on the system bus (e.g., terminate with retry) and on CardBus (e.g., discard read data, terminate write cycles) without informing the card of the snoop results. Therefore, CardBus does not implement these two signals.

   Common silicon which implements **SBO#** and **SDONE,** but not **CBLOCK#,** must strap **SBO#** and **SDONE** high on CardBus PC Cards so the component will always see the CLEAN state. If **CBLOCK#** is implemented, **SBO#** and **SDONE** must be disabled. Strapping them high causes CLEAN to be signaled during the address phase of a write to a locked resource. This causes the component to conclude the write cycle is a writeback of a dirty cache line so it would override the locked status and complete the write. If you intend to implement a function with **SBO#**, **SDONE**, and **LOCK#**, please contact the PCI SIG for assistance in defining this disable mechanism.

3. CardBus does not have the 64-bit bus extension pins.

   Because of pin constraints in the 68-pin connector, CardBus does not support **AD[63::32], C/BE[7::4]#, REQ64#, ACK64#,** and **PAR64**. If common silicon implements these signals, the **REQ64#** signal must be pulled up to **V**CC on CardBus cards so that it is put in 32-bit mode during reset. Floating inputs on **AD[63::32]**, **C/BE[7::4]#**, **PAR64**, and **ACK64#** must be dealt with during runtime. This could be done by enabling input "keepers" when **REQ64#** is seen high during reset.

4. CardBus does not have the JTAG pins.

   Because of pin constraints in the 68-pin connector, CardBus does not support ***IEEE 1149.1 "Standard Test Access Port and Boundary Scan Architecture"*** across the interface. If the capability is not needed in CardBus, common silicon must tie **TRST#** and **TCK** to ground and leave **TDI**, **TDO**, and **TMS** unconnected.

   If the scan chain is to be used on the CardBus card, then the common silicon component and card design must provide a means for manipulating it across the CardBus interface. In general, this will involve incorporating BIST circuitry and using the BIST register defined in configuration space.

5. CardBus has a **CSTSCHG** pin.

   CardBus uses **CSTSCHG** to signal battery low/dead, write protect, ready, and remote wakeup conditions. Common silicon must implement this pin if the functionality is desired for CardBus. It will be a no-connect, or available as a sideband signal when attached to the motherboard, in the PCI environment.

6. CardBus has a **CAUDIO** signal.

   Common silicon requiring access to the speaker must implement this signal. Usage of this silicon in the PCI environment requires either a local speaker, a connection to the system

 ©1999PCMCIA/JEIDA

speaker via a cable from PCI add-in boards, or a sideband signal when mounted on the motherboard.

7. CardBus has a **CCLKRUN#** signal.

   Common silicon desiring a hardware mechanism to start the clock, or continue it for a period of time, must implement this function. **CCLKRUN#** should be strapped low when the PCI environment doesn't support this capability.

8. CardBus only has one interrupt (**CINT#**) pin.

   PCI allows interrupts to be signaled using up to four INTx# pins. Due to pin constraints, CardBus requires using a single interrupt signal, **CINT#**. If common silicon chooses to implement multiple INTx# pins, they must be wire-OR'd together (shorted) when mounted on a CardBus card.

9. CardBus does not provide exceptions to supporting **CPERR#** and **CSERR#**.

   Common silicon must implement the **CPERR#** and **CSERR#** pins along with the associated parity generation and checking logic.

2.1.3.2 Functional Differences

The functional differences between CardBus and PCI are:

1. CardBus doesn't support PCI's Interrupt Acknowledge command.

   This command is intended for the target responsible for interrupt handling duties. Such hardware cannot reside on a CardBus PC Card due to the dynamic insertion/removal nature of PC Cards. A common silicon component, which requires this capability in the PCI environment, must implement this command. This will not cause problems because the command will never occur on CardBus. Also, CardBus will never assign a different meaning to the bit encoding for this command.

2. CardBus doesn't support PCI's Dual address cycle (DAC) command.

   CardBus does not address the issue of 64-bit addressing although the intention is to add it later. Common silicon that could benefit from this command (i.e., bus masters) should implement it. Although the command has no meaning on the CardBus interface based on the present specification, CardBus will never assign a different meaning to the bit encoding for this command.

3. CardBus doesn't address ignoring byte enables when the target determines cacheability.

   In PCI, targets which determine cacheability (e.g., memory controllers) must ignore the byte enables and return all bytes when the access is cacheable. An example for Intel Architecture systems is a memory controller telling the CPU that the access is cacheable via the KEN# signal. This capability does not exist across the CardBus interface.

   This has no impact on common silicon. Even if a function existed, which could signal cacheability information to the CPU and had use in the CardBus environment, the CPU would never use the non-requested bytes returned from CardBus cards. This is because the cacheability indication mechanism will never exit the card.

4. CardBus imposes additional usage rules on **CBLOCK#**.

   CardBus PC Cards, like PCI add-in boards, are targeted for usage across multiple CPU architectures. These differing architectures place different constraints on which transactions can be guaranteed to be atomic. CardBus has additional usage rules for **CBLOCK#** to clarify how exclusive accesses can be guaranteed across differing architectures. These rules apply only to

software developers. The operation of CardBus's **CBLOCK#** pin is identical to PCI's LOCK# pin. Therefore, there is no impact to the design of common silicon.

PCI also requires locking a minimum of 16 bytes, naturally aligned, while CardBus only requires locking what was read. Common silicon must lock at least 16 bytes when lock is established.

5. CardBus does not support PCI's Configuration Mechanism #2.

   PCI requires usage of mechanism #1 for new designs but defined a second mechanism for existing designs. The components which implemented configuration mechanism #2 do not have any use on CardBus PC Cards (e.g., host bridges for PCI). In general, functions which have applicability on PCI and CardBus will never generate configuration cycles.

   However, if common silicon ever does need to generate configuration cycles, it must be designed to use mechanism #1.

6. CardBus requires parity checking and reporting without exception.

   CardBus does not allow PCI's exceptions for functions which "never deal with or contain or access any data which represents permanent or residual system or application state, e.g., human interface and video/audio devices." Therefore, all common silicon must implement the **CPERR#** and **CSERR#** pins along with the associated parity checking and reporting logic for address and data.

7. CardBus's **CSTSCHG** capability requires registers in memory space.

   The implementation of **CSTSCHG** requires providing the **Event**, **Mask**, **Force**, and **Present Value** registers. These registers reside in memory. Common silicon which implements the **CSTSCHG** capabilities for CardBus must implement these registers and indicate the need for address space in the appropriate manner (PCI = base address register in configuration space, CardBus = base address register and the CISTPL_CFTABLE_CB and CS\ISTPL_BAR tuples). A unique base address register is not required for these registers, other memory requirements may be included as appropriate.

8. CardBus does not require cards to drive **CAD[31::0]**, **CC/BE[3::0]#**, and **CPAR** when **CGNT#** is asserted but **REQ#** isn't.

   CardBus doesn't allow cards to be designated the default owners of an idle interface. However, PCI requires any bus master to assume such ownership when directed. Therefore, common silicon must be designed to drive these signals when **CGNT#** is asserted but **CREQ#** isn't. Since this condition will never occur on CardBus, there is no conflict.

2.1.3.3 Electrical Differences

The electrical differences between CardBus and PCI are:

1. CardBus requires special slew rate controlled buffers.

   CardBus cannot tolerate the switching characteristics of PCI buffers due to the limited number of ground pins on the connector. Further, the slew rate controlled buffers typically offered by ASIC houses do not provide enough edge rate control to meet the CardBus requirements. Therefore, common silicon must do one of the following:

   a) Incorporate dual mode buffers (PCI + CardBus modes) and a mechanism to enable the appropriate mode. Note that the primary difference between these two modes is the slew rate control. Therefore, the difference in the buffer's modes could be as simple as reducing or disabling the pre-driver for CardBus.

 ©1999PCMCIA/JEIDA

b) Series terminate each line to get acceptable di/dt characteristics. These resistor must be located very close to the interface component on the card so that signals being driven by the adapter are not affected.

c) Add inductance or resistance to the **VCC** and **GND** paths to pull up or pull down the buffer's supply rails during transitions. This reduces the edge rate by making it a function of the RC time constant of the **VCC**, or **GND**, supply. The rate can be controlled by adjusting the value of resistance/inductance and the amount of current switched through it.

CardBus uses the same DC characteristics as PCI but the loading is reduced. Therefore, input buffer behavior is identical between the two standards. Only in the AC characteristics do differences emerge where CardBus simply specifies a rise/fall time since it operates in a lumped load regime rather than PCI's transmission line environment.

2. CardBus does not support the 5V signaling environment.

   Common silicon must be designed to operate in the 3.3V signaling environment with **VCC**=3.3V. It may additionally be designed to accept **VCC**=5.0V for the PCI domain.

3. Timing differences.

   CardBus is a point-to-point environment so there is no need to distinguish between **CREQ#**, **CGNT#**, and the other signals with respect to timings. The timing differences are:

   a) Tval is measured at the DC Vih and Vil values instead of PCI's 0.4(**VCC**) to cleanly specify what constitutes a valid signal in the CardBus environment.

   b) CardBus specifies the minimum time from **CCLK** stable to the deassertion of **CRST#** as 100 clocks instead of PCI's 100µsec. This change better defines reset requirements with any speed clock. Common silicon must be fully reset in 100 clock cycles.

4. CardBus assigned pins on the connector in a different sequence than PCI.

   This was done to better align CardBus signals with the PCMCIA 2.0 & 1.0/JEIDA 4.1 & 4.0 publication's cards since the adapter must configure the interface for any of the three protocols. The signals affected are **CSERR#**, **CAD16**, **CGNT#**, **CINT#**, **CCLK**, **CRST#**, and **CREQ#**. The impact is to the PC Card or PCI add-in board, since the routing of a common silicon component will require traces crossing each other. The component designer needs to choose the environment in which these crossovers will occur (CardBus vs. PCI).

5. CardBus specifies a maximum current immediately following power up or reset.

   The parameter Icc(CIS) gives CardBus PC Card sockets a guarantee that they can power up any CardBus PC Card and read the CIS without causing a power fault condition by putting too much load on the battery. PCI has no equivalent parameter. Since this may extend beyond the common silicon component, the designer must ensure that cards using their component will not exceed this value when reading the CIS or reading and writing configuration space.

2.1.3.4 Configuration Space Differences

The configuration space differences between CardBus and PCI are:

1. CardBus does not define all the configuration space fields defined by PCI.

   Due to dynamic insertion/removal and generic configuration concerns, CardBus chose to provide all configuration information in the CIS rather than using PCI's predefined header region in configuration space. As a result, CardBus functions must provide configuration information in a different manner than PCI. This means that:

a) Common silicon must implement the 16-byte header required by PCI including Vendor ID, Device ID, Class code, and Revision ID.

b) Common silicon must implement PCI's Max_Lat, Min_GNT, and Interrupt line registers if the function requires them for PCI.

c) The base address registers must be encoded with size information as defined by PCI. These fields, and encodings, are not used by CardBus so there are no implications with respect to common silicon.

2. Certain Command register fields are not optional for CardBus.

   Both PCI and CardBus define a **Command** register in configuration space. CardBus requires common silicon to implement the following fields:

   a) Memory Space must be implemented on all cards which support I/O space because I/O space must be mappable into memory space.

   b) Parity Error Response must always be implemented because CardBus does not allow the exception's provided by PCI.

   c) SERR# Enable must always be implemented because CardBus does not allow the exception's provided by PCI.

3. Certain Status register fields are not optional for CardBus.

   Both PCI and CardBus define a **Status** register in configuration space. CardBus requires common silicon to implement the following fields:

   a) Signaled System Error must always be implemented because CardBus requires all functions to be able to check and report address and data parity errors.

   b) Detected Parity Error must always be implemented because CardBus requires all functions to be able to check and report address and data parity errors.

4. CardBus uses four bytes in the configuration space header for a CIS pointer.

   Common silicon must implement these bytes with a pointer to the beginning of the CIS structure (see the ***Metaformat*** and ***Electrical Specification*** for details). In addition, if the CIS cannot fit in configuration space along with PCI's driver dependent information, common silicon must locate it in memory or expansion ROM space and provide a mechanism(s) to access it (i.e., base address registers and a data path). Unique base address registers are not required.

5. CardBus requires a memory-mapped base address register whenever I/O space is used by the card.

   CardBus requires all I/O space to be mappable into memory space. This means that I/O base address registers must be accompanied with a memory base address register. Therefore, common silicon must implement this companion memory base address register.

2.1.3.5 Software Driver Differences

If a common software driver is to be developed, the following differences must be recognized:

1. CardBus imposes additional interrupt handling requirements.

   CardBus defined the **INTR** field, in the **Event**, **Mask**, **Force**, and **Present Value** registers, so Card Services could do end of interrupt (EOI) processing in environments which don't provide a system-level interrupt handler. This created a standard mechanism for clearing interrupts. Common silicon must implement this field in the above registers. In PCI, the function's device

 ©1999PCMCIA/JEIDA

driver must either use this standard mechanism to clear interrupts or a means to disable it and enable the desired proprietary method must be provided.

2. CardBus requires drivers to have a Card Services interface including tuple comprehension.

   This means that a "common driver" must implement the Card Services interface. When implemented in a PCI system, the driver must first check for its function's existence on PCI. If it doesn't find anything, or if it can handle more than one instance, it must then register with Card Services. If Card Services doesn't exist in the system, determined with the GetCardServicesInfo call, the driver may disable its Card Services interface and proceed knowing no PC Card will show up.

   Upon finding its function, the driver must proceed with completing the configuration in the appropriate manner — tuple traversal for CardBus vs. configuration space interrogation for PCI.

3. Common drivers must handle dynamic insertion and removal events.

   The driver must allow for the fact that its function may disappear or appear at any time and must not treat it as a catastrophic error condition. (See ***4.2: Card–Application Interaction*** guideline for appropriate behavior when this occurs.)

4. PCI specifies that expansion ROM code is never executed in place.

   The only software usage model that PCI defines is where ROM code is always copied from the ROM device to RAM and executed out of RAM. There is nothing in the architecture which prohibits execute-in-place (XIP) operation. On the other hand, CardBus allows this ROM code to be executed-in-place.

   When XIP capability becomes a part of CardBus, a common software driver which wants to take advantage of XIP must deal with differences between the two usage models (copy to RAM for PCI vs. XIP for CardBus). Details of what these differences are cannot be determined until the capability is defined.

© 1999 PCMCIA/JEIDA

2.2 Compatibility Icons

2.2.1 Introduction

Compatibility Icons have been designed for use on both products and product packaging as a standardized method for communicating key product features. By annotating the key compatibility features it becomes a simple process for an end-user to find and purchase compatible products including but not limited to Host Systems and PC Cards.

The Compatibility Icons currently defined for use include:

- The PC Card Logo Denoting PCMCIA or JEIDA Membership
- 16-bit
- CardBus
- 3V Only Operation
- 5V Only Operation
- 3.3V and 5V Operation
- Zoomed Video
- DMA
- DVB
- Thermal Rating Icons

PLEASE CONTACT PCMCIA FOR ARTWORK FILES FOR THE COMPATIBILITY ICONS

2.2.2 Usage Guidelines

All Compatibility Icons utilize the same basic size, shape, and fonts. Although there are some basic deployment guidelines for the icons, the usage of the Compatibility Icons by PCMCIA and JEIDA member companies is completely voluntary.

The following guidelines are provided as a means to improve the end-user recognition and use of the icons when deployed on products and/or product packaging:

- The use of these icons is restricted to member companies of PCMCIA or JEIDA.
- The use of these icons is completely voluntary.
- No new icons may be implemented without the consent of PCMCIA and JEIDA.
- In packaging, documentation, and other literature the original PC Card Logo should be printed first in the series, with the appropriate icons next to it indicating the capabilities of the product.
- For PC Card Sockets and PC Cards, only the compatibility icons should be used.
- PCMCIA suggests that the icons be reproduced no smaller than .25" wide by .15" tall.
- If you have any questions about the usage restrictions or suggestions, please contact the PCMCIA office.

©1999PCMCIA/JEIDA

To assist member companies in determining if/which Compatibility Icons may be of value for their products the following sub-sections are provided with technology overviews and section level cross-references to areas within the ***PC Card Standard***.

2.2.3 Thermal Rating Icons

The Thermal Rating Icons are based upon a Proactive Thermal Management system designed into several of the PC Card Standard component layers. The affected component layers include: software (Card Services and Client Drivers), PC Cards (CIS – Card Information Structure), and Host Systems (HSDT – Host System Data Table).

A high level view of how this integrated thermal management system works is provided as follows:

1. The host system is booted and the system software loads. Card Services retrieves the Host's Thermal Rating value from the HSDT.
2. A PC Card is inserted. Card Services and optional Client Drivers read the card's CIS and retrieve configuration data including the Card's Thermal Rating value.
3. Card Services validates the system resources required for the card's configuration. To validate the Thermal system resource Card Services subtracts the Card's thermal rating value from available Host's thermal rating value.
4. If the result of the subtraction in step 3 is greater than or equal to zero then ample thermal system resource is available and the card configuration process can proceed.
5. If the result of the subtraction in step 3 is less than zero than the card requires more thermal resource than the host currently has available so the card can not be configured. Enhanced system software working with Card Services may provide the user feedback indicating that the card could not be configured as well as offering possible user actions or options to restart the process.

   Note: In a multi-slot host configuration the slots may share the system's thermal resources. Hence, a second card may need less than the system's rating value but still not be configurable due to another card already being present and using some of the thermal resource.

As previously noted, cases can arise where cards may not be configurable due to either the capabilities of the host (e.g., laptop versus palmtop) or the presence of other cards. To help identify these conditions well before they occur (e.g., at purchase time) manufacturers can use the thermal rating icons on their packaging and the customer can perform the subtraction logic to determine if the desired products are compatible. To assist with the customer usage of the Thermal Rating Icons two slightly different icon formats have been defined.

The thermal icon format defined for PC Cards is presented as follows:

Figure 2-1: Card-Side Thermal Icons

The thermal icon format defined for host systems is presented as follows:

Figure 2-2: Host-Side Thermal Icons

Both formats utilize a unitless whole number value provided in increments ranging from 1 to 99. The Thermal Rating Icon values are based directly off of the CIS/HSDT values. The CIS/HSDT values are converted to whole numbers by multiplying them by 10 and then rounding to the nearest whole number. For example, a CIS/HSDT value of 1.23 would be converted to a reported icon value of 12. A CIS/HSDT value of 0.75 would be a reported icon value of 8.

For more information regarding the technical details of the various thermal management components please refer to the following volumes/sections of the ***PC Card Standard***:

- For PC Card Thermal Ratings details see the ***Physical Specification*** – PC Card Thermal Ratings section.
- For CIS Thermal tuple data see the ***Metaformat Specification*** – CISTPL_CFTABLE_ENTRY, TPCE_MI Field and CISTPL_CFTABLE_ENTRY_CB, TPCE_CBMI Field description sections.
- For System Software and Driver details see the ***Card Services Specification*** – ConfigureFunction and InquireConfiguration service description sections.
- For HSDT and Host Thermal Ratings details see the ***PC Card Host System Specification***.

©1999PCMCIA/JEIDA

3. PHYSICAL GUIDELINES

UNLESS OTHERWISE SPECIFIED, ALL DIMENSIONS ARE IN MILLIMETERS (MM).

3.1 15 Position Shielded Latching I/O Connector

3.1.1 I/O Connector

This section specifies physical characteristics of Open System LAN and modem I/O connectors. Open System PC Cards are those which contain I/O connectors compliant with this section, including keying assignments and are further compliant with functional and electrical descriptions.

Open System compliance:

- Does not guarantee interoperability of independent implementations.
- Does not guarantee certifiability to any external (non-PCMCIA/JEIDA) organizations including FCC, VDE, IEC, UL, etc.
- Does guarantee electricial damage avoidance associated with inadvertent connection.
- Does facilitate uniform implementation interfaces for developers.
- Does promote availability of multi-sourced I/O connectors satisfying PC Card Standard-mandated design criteria.

PC Cards with I/O connectors or cable assemblies which are not compliant with the Open System are Closed System cards. This specification does not define nor address Closed System cards.

© 1999 PCMCIA/JEIDA

3.1.1.1 Card I/O Connector

The male PCB connector shall be configured as shown in Figure 3-1: PCB Connector and Table 3-1: PCB Connector Dimensions and Table 3-2: PCB Connector Contact Lengths.

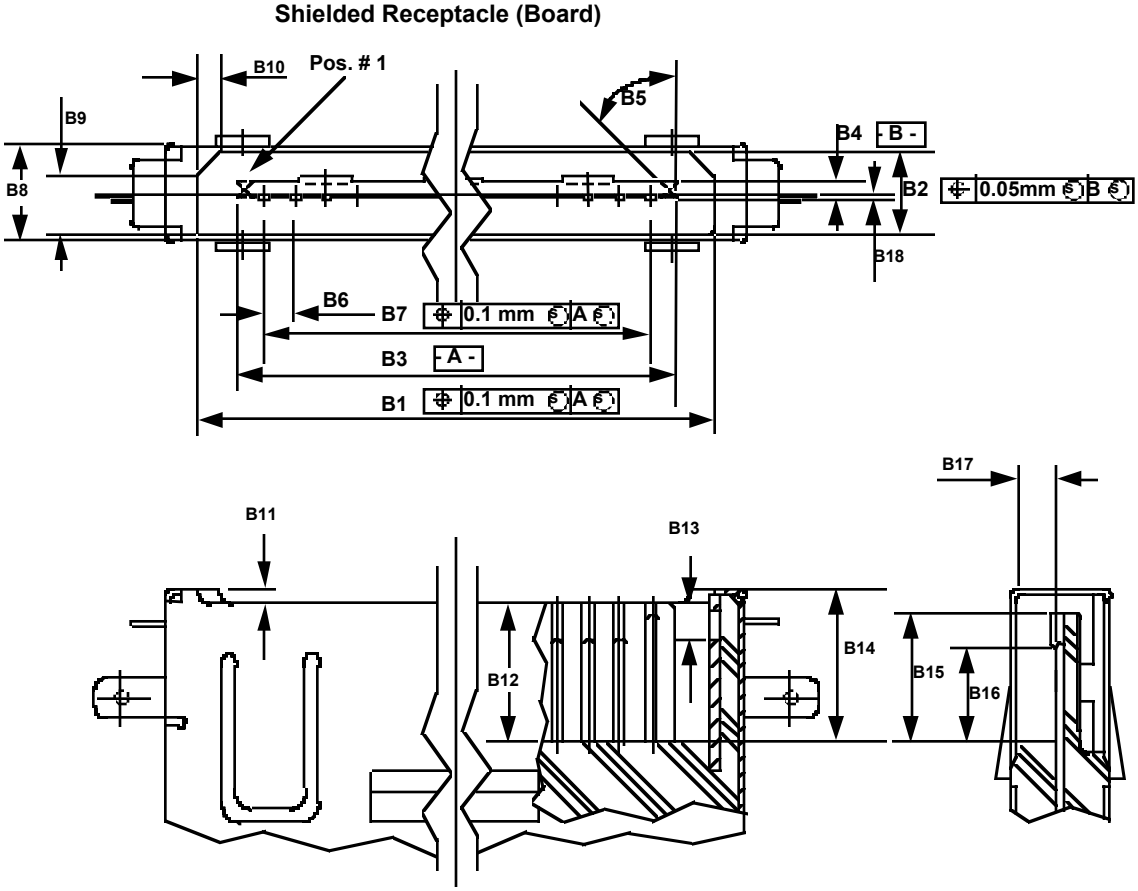


Figure 3-1: PCB Connector

©1999PCMCIA/JEIDA

Table 3-1: PCB Connector Dimensions

Dimension	Millimeters
B1	14.8 ± 0.15
B2	2.5 +0, -0.1
B3	12.38 +0, -0.08
B4	0.7 ± 0.05
B5	30° +0.5, -0
B6	0.8 ±0.05
B7	11.2 ±0.1
B8	2.9 Max
B9	2.0 ±0.1
B10	0.8 ±0.1
B11	0.5 Ref
B12	3.9 Min
B13	1.2 Ref
B14	4.5
B15	4
B16	Refer toTable 3-2
B17	0.95 ±0.1
B18	0.2 ±0.3

Table 3-2: PCB Connector Contact Lengths

Connector Pin Numbers	Contact Length Millimeters
Pins 1,15	3.9
Pins 2..14	3.0

© 1999 PCMCIA/JEIDA

3.1.1.2 Cable I/O Connector

The female cable plug connector shall be configured as shown in the figure below and Table 3-3: Cable Connector Dimensions

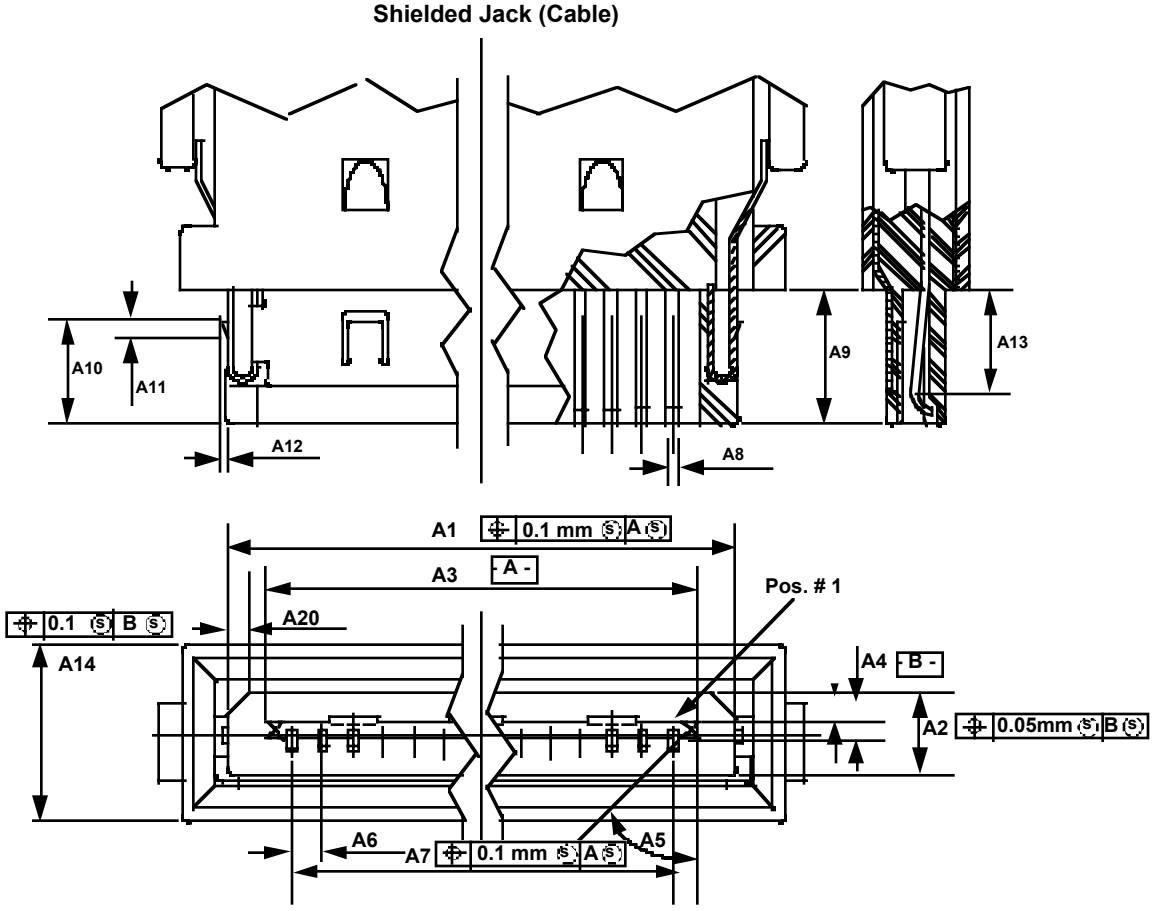


Figure 3-2: Cable Connector

©1999PCMCIA/JEIDA

Table 3-3: Cable Connector Dimensions

Dimension	Millimeters
A1	14.6 Max
A2	2.4 Max
A3	12.4 Min
A4	0.8 Min
A5	30° +0, -0.05
A6	0.8 ±0.05
A7	11.2 ±0.1
A8	0.3 ±0.05 Ref
A9	4.5 Max
A10	3.7 ±0.2
A11	0.9 Max
A12	0.25 Ref
A13	3.0 Ref
A14	6.0 Max

3.1.1.3 Key Standards

The dimensions and locations of keys are shown in the figure below and in Table 3-4: Key Dimensions.

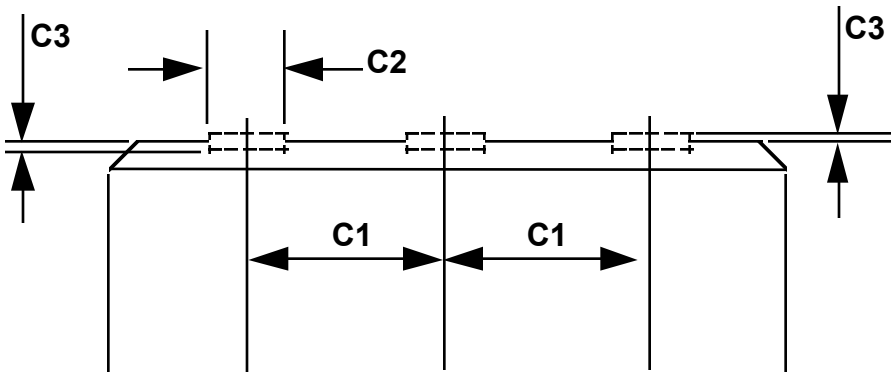


Figure 3-3: Key Size and Locations

Table 3-4: Key Dimensions

Dimension	Millimeters
C1	4 ±0.1
C2	1.6 ±0.1
C3	0.25±0.05

© 1999 PCMCIA/JEIDA

Three keying configurations are defined, one for LAN and two are reserved for future definition. Figure 3-4 shows LAN keying.

Figure 3-5 and Figure 3-6 show those reserved for future definition.

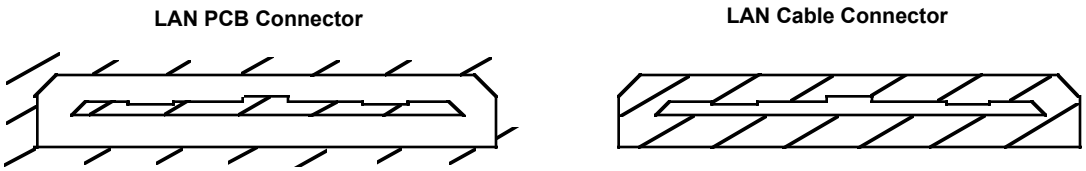


Figure 3-4: Open Standard LAN Connector Key Configuration

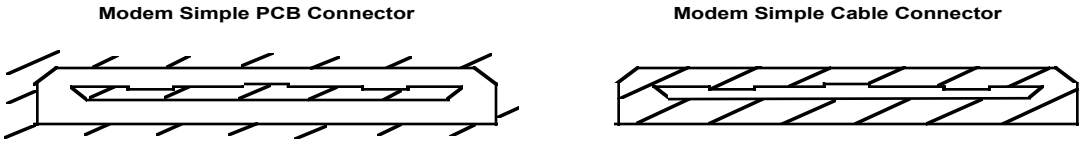


Figure 3-5: Reserved Connector Key Configuation

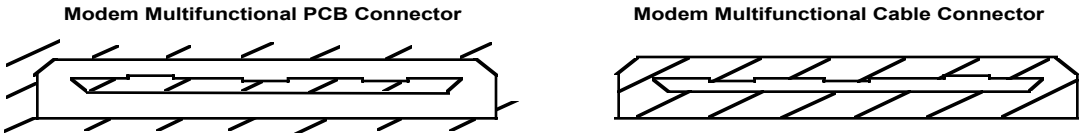


Figure 3-6: Reserved Connector Key Configuration

3.1.1.4 Plating

The outermost plating of the connector system contact mating area shall be hardened gold, or materials compatible with gold.

The outermost plating of the shield shall be nickel plated or other plating material compatible with nickel. Recommended shield material is stainless steel.

The interconnect system shall pass all requirements of Section 3.1.2, I/O Connector Reliability and Section 3.1.3, Connector Durability.

3.1.1.5 Test Sequence

The following table defines recommended test sequences. The table assumes 12 test groups, each represented by a vertical column. One or tests are performed on each group. In a particular column, the number 1 appears in the row of the test to be executed first, the number 2 in the row of the test to be executed second and so on.

©1999PCMCIA/JEIDA

Table 3-5: Recommended Test Sequence

Test No	Test item	Test group												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Test sequence														
1	Insulation Resistance					1,7	1							
2	Dielectric Withstanding Voltage					2	2							
3	Contact Resistance (low level)		1,5	1,4,6,8,10	1,4	3,6	3	1, 5	1, 3	1	1,3			
4	Contact Forces		2	2				2						
5	Connector Insertion/withdrawal Forces		3	3				3						
6	Contact Retention Force	1												
7	Durability-Office Environment		4											2
8	-Harsh Environment			5										
9	Vibration				2									
10	Mechanical Shock				3									
11	Humidity (Normal Condition)			7		4								
12	Moisture Resistance						4							
13	Thermal Shock					5								
14	High Termperature Life							4						
15	Cold Resistance								2					
16	Hydrogen Sulfide			9						2				
17	Salt Water Spray										2			
18	Current Rating											1		
19	Latch Strength		6											
20	Mechanical Torque/Flex												1	
21	Polarization & Key Force													1,3

3.1.2 I/O Connector Reliability

The I/O interconnect system as specified in Section 3.1.1 shall meet or exceed all reliability test requirements of this Section. Unless otherwise specified, all test and measurements shall be made at:

Temperature	15°C to 35°C
Air pressure	86 to 106 kPa
Relative humidity	25% to 85 %

If conditions must be closely controlled in order to obtain reproducible results, the parameters shall be:

Temperature	23°C ± 1°C
Air pressure	86 to 106 kPa
Relative humidity	50% ± 2%

© 1999 PCMCIA/JEIDA

3.1.2.1 Mechanical Performance

The I/O interconnect system mechanical performance is specified as follows:

3.1.2.1.1 Office Environment

Standard	Testing
Guaranteed number of insertions/extractions = 10,000 min.	Paragraph 3.1.3.1

3.1.2.1.2 Harsh Environment

Standard	Testing
Guaranteed number of insertions/extractions = 5,000 min.	Paragraph 3.1.3.2

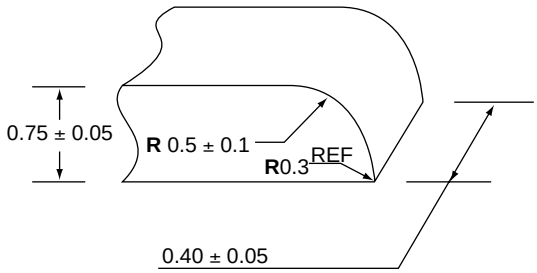
3.1.2.1.3 Total Insertion Force (with latches disengaged)

Standard	Testing
29.4 N6.6 lbs. (3.0 kg) max.	Insert at speed of 25.4mm/min

3.1.2.1.4 Total Pulling Force (with latches disengaged)

Standard	Testing
1.5 lbs. (0.68 kg) min.	Extract at speed of 25.4mm/min

3.1.2.1.5 Single Contact Forces

Standard	Testing
Using gauge pin shown below: 0.220 lbs. 100 grams max insertion force 0.044 lbs. 20 grams min. extraction force  Tool steel gauge pin (HRC=50-55, surface finish= .406μmm max, chamfer all sharp edges)	Insert and extract gauge pin at speed of 25.4mm/min

3.1.2.1.6 Single Contact Holding Force

Standard	Testing
Contact shall not be displaced from insulator when a 1 kg minimum force is applied to the contact in the axis of retention.	Apply a minimum force of 1 kg to the contact in both directions along the axis of retention while holding the insulator rigid.

©1999PCMCIA/JEIDA

3.1.2.1.7 Vibration

Standard	Testing
a. No mechanical damage shall occur on the parts.	MIL-STD-202F, Method 204D
b. No current interruption greater than 100 nsec.	Test condition B (15 G peak)
c. Contact resistance 3.1.2.2.1b.	10 Hz to 2000 Hz

3.1.2.1.8 Shock

Standard	Testing
a. No mechanical damage shall occur on the parts. b. No current interruption greater than 100 nsec.	MIL-STD-202F, Method 213B Acceleration 50G, Standard holding time 6 msec, Semi-sine wave.

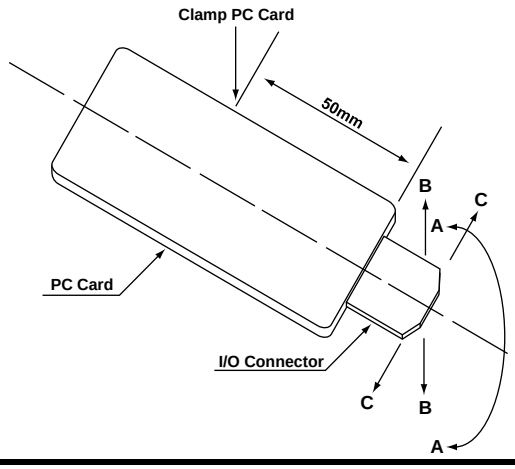
3.1.2.1.9 Latch Retention Force

Standard	Testing
12 lbs. (5.5 kg) min	Minimum latch retention force before and after durability (office environment) testing.

3.1.2.1.10 Polarization and Key Force

Standard	Testing
13 lbs. (6 kg) minimum, polarization and keys must remain intact and functionally to this requirement after durability testing.	Connector system must not mate when keyed cable plug connector is mismated upside down to PCB male connector, or with any other polarization Type, or any other keying standard, with a force up to 13 lbs.

3.1.2.1.11 Connector Plug Torque and Flex

Standard	Testing
a. No mechanical damage shall occur on the parts. b. No current interruption greater than 100 nsec. 	1) Clamp PC Card at I/O connector end, within 50 mm of the card edge~~15 pin connector end~~. 2) Apply clockwise torque of 0.226 N-m to the cable end of the connector, hold at extreme for 5 minutes. Repeat test in counter-clockwise direction. Repeat 5 times in each direction. 3) Apply upward torque of 0.226 N-m to the cable end of the connector. Hold for 5 minutes. Repeat in downward, left and right directions. Repeat 5 times in all 4 directions. 4) Apply upward torque of 0.0226 N-m to the cable end of the connector. Hold for 3 seconds. Repeat in downward, left and right directions. Repeat 1000~~10,000~~ times in all 4 directions.

© 1999 PCMCIA/JEIDA

3.1.2.1.12 Strain Relief

Standard	Testing
The strain relief shall be capable of retaining the attaching cable with no separation or damage to conductor(s), shielding (if applicable), or insulation when subjected to force equal or greater than that required to extract the connector.	Clamp connector housing. Apply force to cable outward from connector housing. If cable assembly does not implement latching mechanism, force shall be 1.6 Kg. If cable assembly implements latching mechanism, force shall be 6.4 Kg or the maximum observed during the tests of paragraph 3.1.2.1.9.

3.1.2.2 Electrical Performance

The I/O interconnect system electrical performance is specified as follows.

3.1.2.2.1 Contact Resistance (low level)

Standard	Testing
a. Initial.....40 mΩ maximum b. After test...45 mΩ maximum	MIL-STD-1344A, METHOD 3002.1 Open voltage 20 mV Test current 1 mA a) Measure and record the initial resistance (R_i) of the connector system (from attachment of male connector to the PCB, to the wire of the cable plug connector): $R_i \leq 40$ mΩ b) Measure and record resistance (R_f) of the connector system after test of the connector system. Resistance value after test: $R_f \leq 45$ mΩ

3.1.2.2.2 Withstanding Voltage and Isolation Voltage

Standard	Testing
a) No shorting, breakdown, flashover or other damages when 1000 Vrms AC is applied for 1 minute between adjacent contacts or from each contact to shell. b) Current leakage 1 mA max.	MIL-STD-202F, METHOD 301, measured between adjacent contacts, and measured from contact to shell

3.1.2.2.3 Insulation Resistance

Standard	Testing
a. Initial.....1,000 MΩ min. b. After test.....100 MΩ min.	MIL-STD-202F, METHOD 301 Measure within 1 minute after applying 500 V DC

©1999PCMCIA/JEIDA

3.1.2.2.4 Current Capacity

Standard	Testing
0.5 A per contact	20 °C max temp rise over ambient at rated current

3.1.2.2.5 Insulation Material

Standard	Testing
UL 94 V-0 rated material in thickness used	Per UL Specification 94

3.1.2.3 Environmental Performance

3.1.2.3.1 Operating Environment

Standard
Operating Temperature: -20 °C to + 60 °C Relative Humidity: 95% Max. (non-condensing)

3.1.2.3.2 Storage Environment

Standard
Operating Temperature: -20 °C to + 70 °C Relative Humidity: 95% Max. (non-condensing)

3.1.2.4 Environmental Resistance

3.1.2.4.1 Moisture Resistance

Standard	Testing
Contact resistance: 3.1.2.2.1 Insulation resistance: 3.1.2.2.3 b.	MIL-STD-202F, METHOD 106E (excluding vibration) 10 cycles (1 cycle=24 hours) with connector system mated

3.1.2.4.2 Thermal Shock

Standard	Testing
No physical damage shall occur during testing Contact resistance: 3.1.2.2.1 Insulation resistance: 3.1.2.2.3 b.	MIL-STD-202F, METHOD 107G Test condition A, -55 °C to + 70 °C 5 cycles (1 cycle=1 hours) with connector system mated

3.1.2.4.3 Durability (High Temperature)

Standard	Testing
Contact resistance: 3.1.2.2.1 b.	MIL-STD-202F, METHOD 108A Test Condition B, 70 °C, 250 hours min. with connector system mated

© 1999 PCMCIA/JEIDA

3.1.2.4.4 Humidity (Normal Condition)

Standard	Testing
Contact resistance: 3.1.2.2.1 Insulation resistance: 3.1.2.2.3 b .	MIL-STD-202F, METHOD 103B Test Condition B, 40 ^{o}C , 90 to 95% RH with connector system mated

3.1.2.4.5 Mixed Flowing Gases

Standard	Testing
No visible corrosion under 3X magnification Contact resistance: 3.1.2.2.1 except max. increase to be 20 mΩ above initial measurement	EIA-364-65 (TP-65), Class II environment for 48 hours after stablization. Performed with 50% of test group connectors mated; rest unmated. Measurement made on 100% of contacts on all connectors in test group.

3.1.2.4.6 Salt Water Spray

Standard	Testing
No harmful corrosion (or degradation of contact performance) should occur on the contacts. Contact resistance 3.1.2.2.1	MIL-STD-202F, METHOD 101D Test condition B, Concentration 5% 35 ^{o}C, 48 hours min., with connectors unmated

3.1.3 Connector Durability

The I/O interconnect system as specified in Section 3.1.1 shall meet or exceed all durability requirements of this subsection.

Test conditions for the mate/unmate cycles are:

Cycle rate	400-600 cycles per hour
Temperature	15 ^{o}C to 35 ^{o}C
Air pressure	86 to 106 kPa
Relative humidity	25% to 85%

3.1.3.1 Office Environment

The office environment is defined in EIA-364A as class 1.1 - year round air conditioning (non-filtered) with humidity control.

Test sequence:

Contact resistance per 3.1.2.2.1 a.
Mate and unmate the connector for a total of 10,000 cycles
Contact resistance per 3.1.2.2.1 b.

©1999PCMCIA/JEIDA

3.1.3.2 Harsh Environment

The harsh environment is defined in EIA-364A as class 1.3 — no air conditioning, no humidity with normal heating and ventilation.

Test sequence:

Contact resistance per 3.1.2.2.1. a.	
Mate and unmate the connector 1,000 cycles	total cycles = 1,000 cycles
Humidity per 3.1.2.4.4	
Mate and unmate the connector 1,000 cycles	total cycles = 2,000 cycles
Humidity per 3.1.2.4.4	
Mate and unmate the connector 3,000 cycles	total cycles = 5,000 cycles
Humidity per 3.1.2.4.4	
Mixed Flowing Gases per 3.1.2.4.5	

3.1.4 PC Card LAN 15 Position I/O Connectivity

This recommendation defines the pinout for LAN PC Cards. Several specific pinouts are defined, each serving a different technology or technology-media combination. All assume the use the same physical connector and keying shown for LAN in Figure 3-1, Figure 3-2, and Figure 3-3. All Open System LAN PC Cards use the common 15 pin I/O connector. The same swap cable assembly works with two popular network configurations, Ethernet 10BASE-T and Token Ring UTP. Pin assignments are chosen so that accidental connection of an open standard LAN card to a different technology network will not cause damage to the card or the network.

The option is provided to power external transcievers through the PC Card. This simplifies some configurations. The option is also provided to power the PC Card from an external supply allowing LAN cards to be utilized in very low power platforms. Two status LEDs can be powered through the cable assembly.

© 1999 PCMCIA/JEIDA

3.1.4.1 Connector Pinout Configurations

The pinouts defined for Open System LAN I/O Connector are shown in the following tables.

Table 3-6: Ethernet with AUI and 10BaseT

Pin	Signal Description
1	10BASE-T receive, negative phase
2	10BASE-T receive, positive phase
3	Power into PC card from external supply
4	AUI receive, negative phase
5	AUI receive, positive phase
6	10BASE-T link status LED
7	Cable activity (receive) status LED
8	AUI transmit, negative phase
9	AUI transmit, positive phase
10	Power out of PC card to external modules
11	AUI collision input negative phase
12	AUI collision input positive phase
13	Ground
14	10BASE-T transmit, negative phase
15	10BASE-T transmit, positive phase

Table 3-7: Ethernet with AUI and 10 Base 2

Pin	Signal Description
1	Not connected
2	Not connected
3	Power into PC card from external supply
4	AUI receive, negative phase
5	AUI receive, positive phase
6	Transmit status LED
7	Cable activity (receive) status LED
8	AUI transmit, negative phase
9	AUI transmit, positive phase
10	Power out of PC card to external modules
11	AUI collision input negative phase
12	AUI collision input positive phase
13	Ground
14	10BASE-2 coax shield
15	10BASE-2 coax center conductor

©1999PCMCIA/JEIDA

Table 3-8: Token Ring UTP and STP

Pin	Signal Description
1	UTP transmit, negative phase
2	UTP transmit, positive phase
3	Power into PC card from external supply
4	STP receive, negative phase
5	STP receive, positive phase
6	Transmit status LED
7	Receive status LED
8	Not connected
9	Not connected
10	Power out of PC card to external modules
11	UTP receive, negative phase
12	UTP receive, positive phase
13	Ground
14	STP transmit, negative phase
15	STP transmit, positive phase

Table 3-9: ARCNET with UTP and Coax

Pin	Signal Description
1	Not connected
2	Not connected
3	Power into PC card from external supply
4	Not connected
5	Not connected
6	Transmit status LED
7	Cable activity (receive) status LED
8	UTP negative phase
9	UTP positive phase
10	Power out of PC card to external modules
11	Not connected
12	Not connected
13	Ground
14	Coax shield
15	Coax center conductor

© 1999 PCMCIA/JEIDA

Table 3-10: ATM

Pin	Signal Description
1	UTP transmit, negative phase
2	UTP transmit, positive phase
3	Power into PC card from external supply
4	External transmit, negative phase
5	External transmit, positive phase
6	link status LED
7	Cable activity (receive) status LED
8	External receive, negative phase
9	External receive, positive phase
10	Power out of PC card to external modules
11	External Signal Detect, negative phase
12	External Signal Detect, positive phase
13	Ground
14	UTP receive, negative phase
15	UTP receive, positive phase

3.1.4.2 Common Mode Note

There is an issue which must be considered regarding pins 11, 12. It is physically possible via a standard swap cable to connect these, accidentally, to a Token Ring MAU. The IEEE 802 specification indicates that such a cable connection must be able to sustain high common mode voltages without damage. Solutions to this problem are implementation specific. One solution is to provide an isolated receiver as is in 802.3 ethernet coax transceivers. Since this is only a receiver, the power requirements are very low and implementations should be low cost.

3.1.4.3 Cable Assembly Notes

The PC Card LAN pinout assignments specified in this document allow for several levels of functionality in cable assemblies. For example, the cable assembly shown in the following table would connect any Pinout Configuration 1 PC card to an Ethernet 10BASE-T concentrator ***or*** any Pinout Configuration 3 PC Card to a Token Ring MAU.

PC Card Connector Pin		RJ45 Connector Pin
1	Twisted	3
2	Pair	6
11	Twisted	4
12	Pair	5
14	Twisted	1
15	Pair	2

©1999PCMCIA/JEIDA

Adding LEDs to the external box or molding which houses the RJ45 connector and utilizing PC Card connector pins 6, 7 and 13 according to the following table adds additional value to the cable assembly.

PC Card Connector Pin		External Module
6	Single Conductor	Link/Transmit LED
7	Single Conductor	Receive LED
13	Single Conductor	Ground Returns From Both LEDs

Utilizing PC Card pins 3, 10 and 13 adds value to the cable assembly by allowing for power from the PC Card to an external module (for example, to power an Ethernet AUI to 10BASE-2 transceiver) and/or power from an external power source into the PC Card (for example, to power the PC Card in a low power palmtop PC). Connections for this are shown in the following table.

PC CARD CONNECTOR PIN		EXTERNAL MODULE
3	Single Conductor	External Module Power Input
10	Single Conductor	External Power Source
13	Single Conductor	Ground

© 1999 PCMCIA/JEIDA

3.2 Modem I/O Unshielded Connector for Open Systems

3.2.1 Introduction

3.2.1.1 Purpose

This document describes connectors suitable for use as external input/output interface from PC Cards used for Modem, FAX, voice and audio applications in accordance with applicable PC Card standards. Applicable cards are those which adhere to the ***Open Standard for I/O Interconnection***. Applications and testing are based upon "harsh environments."

3.2.1.2 Scope

3.2.1.2.1 Modem/FAX Cards

The intent of this document is to provide sufficient information for Modem/FAX, I/O Card Developers to design such devices for use with external media access devices by using common connectors and interface. This document describes connectors suitable for use with Type II and Type III Modem/FAX PC Cards incorporating internal DAAs. Also supported are Modem/FAX PC Cards incorporating audio features which interface to external audio devices having speaker and microphone capabilities. It is intended to allow such Modem/FAX PC Cards to use cables involving connectors described herein, interchangeably without concern for physical or electrical damage, when properly applied. PC Card Standard specified Signal-Contact interconnection assignments shall be according to Section 3.2.7 herein.

3.2.1.2.2 Connectors

This document is intended to provide sufficient information of applicable mechanical and interface parameters involved with the use and selection of the connectors described herein. This specification covers only information that is applicable to the mating interface and is not intended to be a complete product specification.

©1999PCMCIA/JEIDA

3.2.2 Overview

The potential proliferation of modems and specialized communications cards, all based upon the ***PC Card Standard***, has lead to a need for standardization on connectors for direct connection between the PC Card and various media.

In order to assure that any I/O Card could have any access devices connected without harm, the Open Connector concept allows for the definition of interface signal levels and mechanical parameters to assure that different PC Card form factors are not subject to damage when properly terminated with the connectors herein.

It is accepted that vendors may wish to have different interconnection schemes that would function properly only when interconnect components of their specific choice are used. These are termed "closed systems." Interchangeability with respect to cabling and the mechanical I/O interface of the modem cards (per this specification) is not expected nor desired. Vendors using such "closed systems" are instructed to not use the connectors described in this specification so that potentially destructive interface problems will be prevented.

Since PC Cards used for Modem applications may be used in non-office environments as part of mobile operating systems, testing herein is based upon standards for "harsh environments" as defined by the ***PC Card Standard***.

This specification gives several options to designers and users of PC Card modems. The "minimum interface" presents a standardization of a 4 position connector for applications wherein only one type of simple interface is desired. The "maximum interface" connector provides a split-insert, in two sections, with 4 and 3 contacts respectively. This maximum interface connector can receive a cable with a 4 position plug for one type of interface, or it can receive a separate cable with a 3 position plug for another type of interface. If a more complicated interconnection is required, a third cable having 7-positions may be used. It is expected that all ***PC Card Standard*** compliant cards for Modem/FAX applications will have either a 4 or 7 position card connector installed at the end of the card opposite that which is inserted into the host's socket.

Due to expected structural constraints of applicable PC Card modems, the connectors defined herein are suitable for all PC Cards except the thinner Type I. Wherever applicable, illustrations and testing discussed in this specification shall incorporate Type II PC Cards.

Current Modem and Modem/FAX Card applications do not require use of shielded cables. Therefore, for economical benefits, connectors described herein are not intended for use with shielded cables or shell-to-case grounds. Connectors for Local Area Network applications may need to be shielded and may be described by other PC Card specifications.

The "open system" compliance presented herein does not guarantee interoperability of independent implementation schemes, nor does it guarantee certifiability to any external (non-PCMCIA/JEIDA) criteria such as FCC, UL or the like. However, it does prevent mechanical damage associated with inadvertent interconnection. "Open system interconnection" facilitates uniform interfaces for PC Card Developers by promoting availability of multi-sourced I/O connectors satisfying the ***PC Card Standard's*** criteria per this specification.

© 1999 PCMCIA/JEIDA

3.2.3 Connector Physical

3.2.3.1 Card I/O Connectors

The "minimum interface" 4 position connector is shown in ***Figure 3-7: Card I/O Connector, 4 Position*** with dimensions per ***Table 3-11: Card I/O Connector Dimensions***. The "maximum interface" 7 position connector is shown in ***Figure 3-8: Card I/O Connector, 7 Position*** with dimensions also per ***Table 3-11***.

Card I/O Connector, 4 position per ***Figure 3-7: Card I/O Connector, 4 Position***.

Card I/O Connector, 7 position per ***Figure 3-8: Card I/O Connector, 7 Position***.

Card I/O Connector dimensions per ***Figure 3-11: Cable Plug Connector, 7 Position***.

3.2.3.2 Cable Plug Connectors

3.2.3.2.1 Identification

Cable Plug Connector, 3 position per ***Figure 3-9: Cable Plug Connector, 3 Position***.

Cable Plug Connector, 4 position per ***Figure 3-10: Cable Plug Connector, 4 Position***.

Cable Plug Connector, 7 position per ***Figure 3-11: Cable Plug Connector, 7 Position***.

Cable Plug Connector dimensions per ***Table 3-12: Cable I/O Connector Dimensions***.

3.2.3.2.2 Reservations

Use of the 7 position and 3 position cable plugs with the 7 position card I/O receptacle are reserved for future circuit definition(s).

©1999PCMCIA/JEIDA

3.2.3.2.3 Card I/O Connector

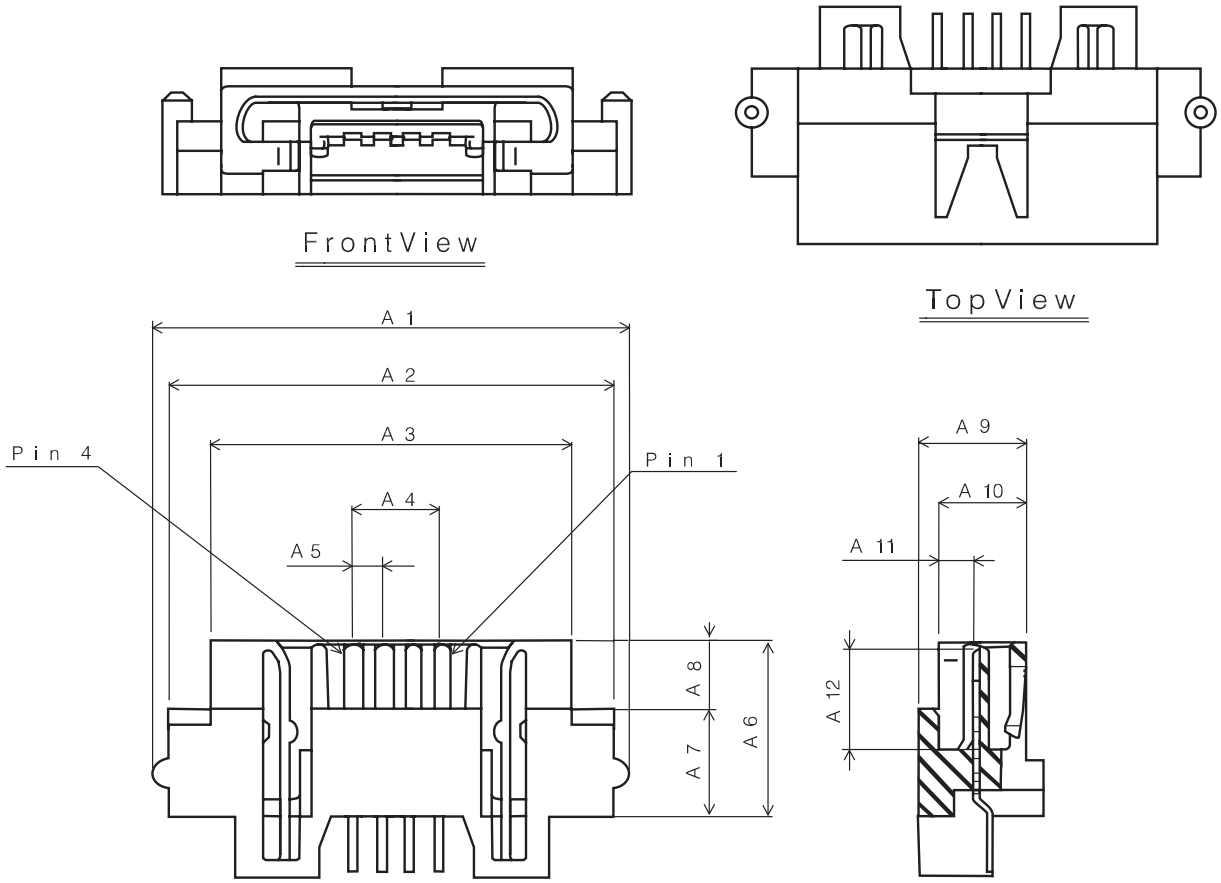


Figure 3-7: Card I/O Connector, 4 Position

Table 3-11: Card I/O Connector Dimensions

Dimension	4 Position Millimeters		Inches		7 Position Millimeters		Inches	
A1	16.4	± 0.2	.646	± .008	24	± 0.2	.945	± .008
A2	15.4	± 0.2	.606	± .008	23	± 0.2	.906	± .008
A3	12.4	+0 -0.1	.488	+0 -.004	20	+0 -0.1	.787	+0 -.004
A4	3	± 0.15	.118	± .006	11	± 0.15	.433	± .006
A5	1	± 0.05	.039	± .002	1	± 0.05	.039	± .002
A6	(6)		(.236)		(6)		(.236)	
A7	3.7	± 0.2	.146	± .008	3.7	± 0.2	.146	± .008
A8	2.3	± 0.1	.090	± .004	2.3	± 0.1	.090	± .004
A9	3.7	+0 -0.1	.146	+ 0 -.004	3.7	+ 0 -0.1	.146	+ 0 -.004
A10	3	+ 0 -0.1	.118	+ 0 -.004	3	+ 0 -0.1	.118	+ 0 -.004
A11	1.2	± 0.1	.047	± .004	1.2	± 0.1	.047	± .004
A12	3.45	± 0.1	.136	± .004	3.45	± 0.1	.136	± .004

Note: Figures in brackets ( ) are “reference.”

© 1999 PCMCIA/JEIDA

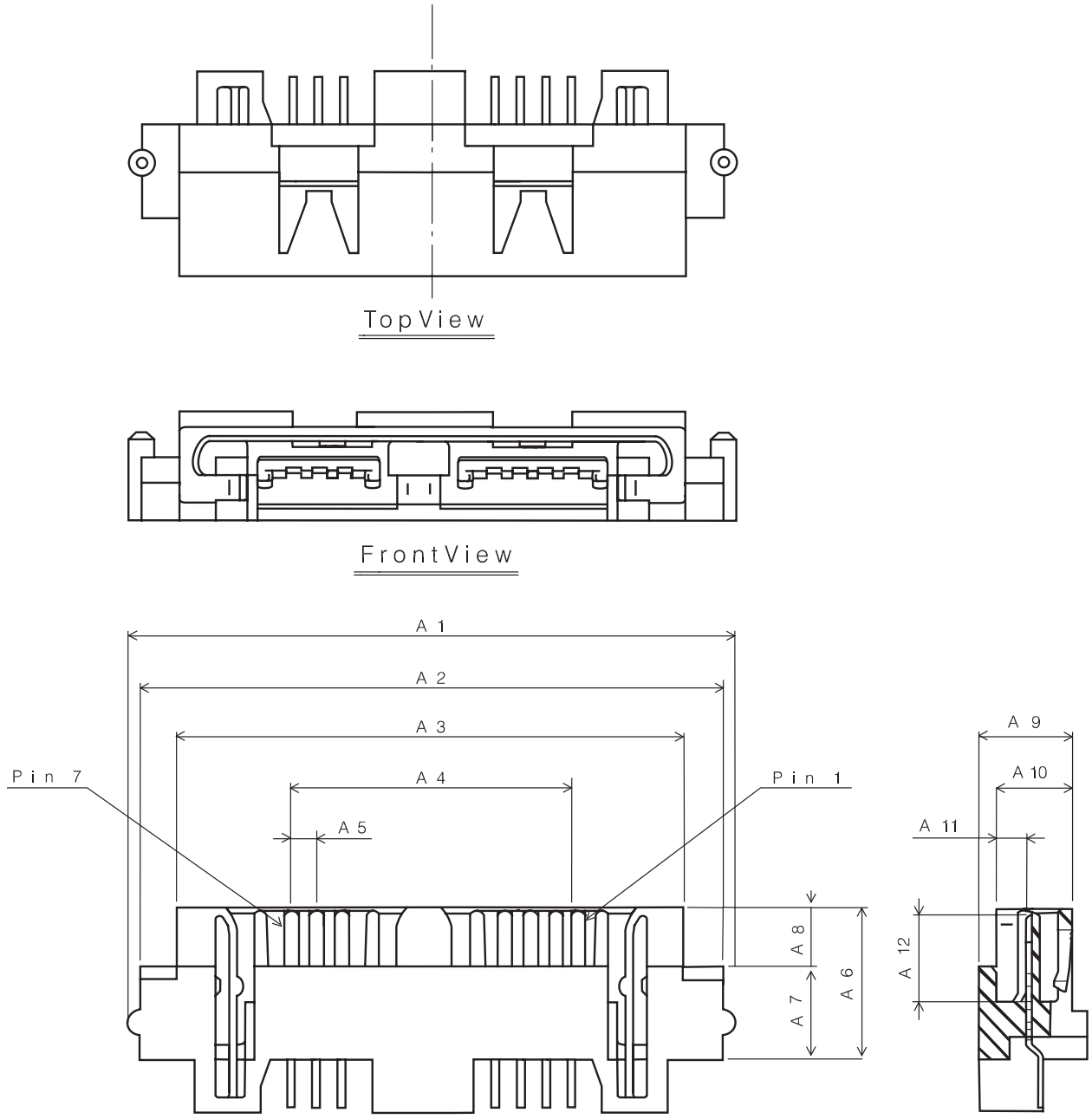


Figure 3-8: Card I/O Connector, 7 Position

©1999PCMCIA/JEIDA

3.2.3.2.4 Cable Plug Connector

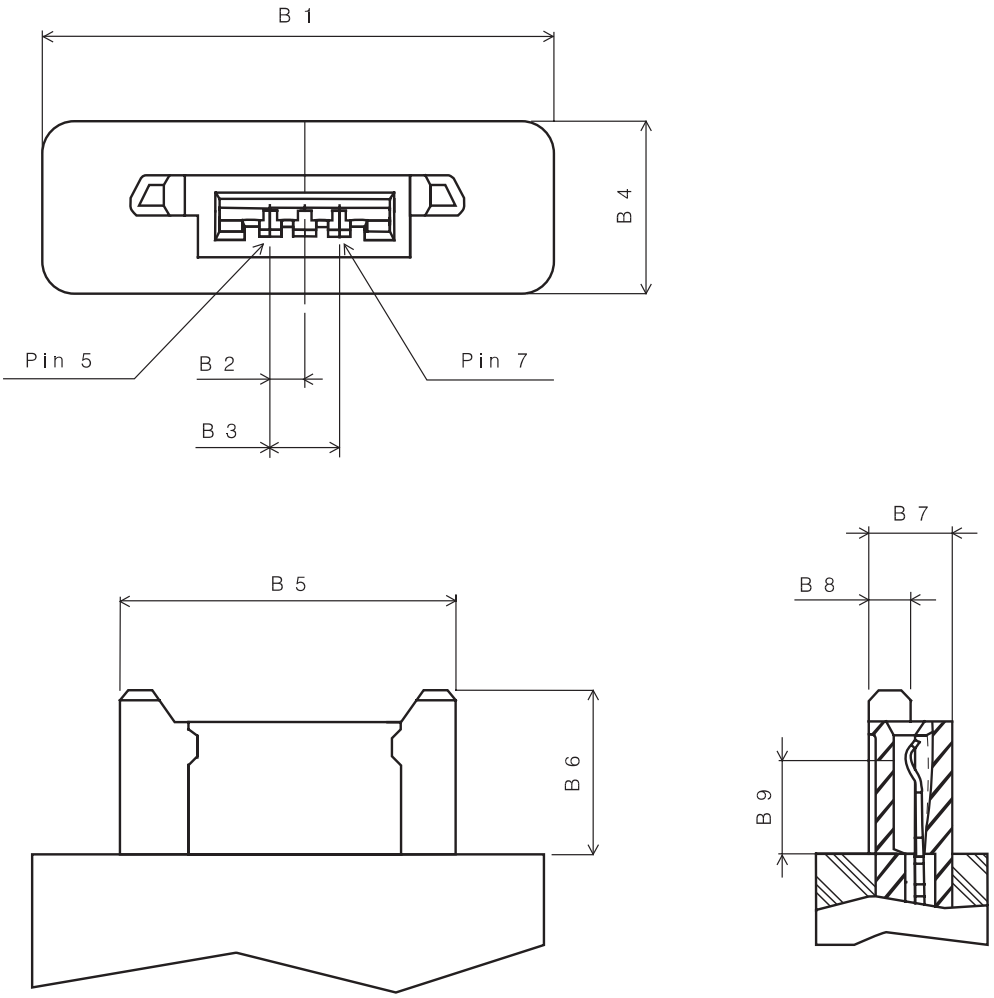


Figure 3-9: Cable Plug Connector, 3 Position

© 1999 PCMCIA/JEIDA

Table 3-12: Cable I/O Connector Dimensions

Dimension	3 Position Millimeters		Inches		4 Position Millimeters		Inches		7 Position Millimeters		Inches	
B1	15.0	± 0.3	.590	± .012	15.0	± 0.3	.590	± .012	24.5	± 0.3	.590	± .012
B2	1.0	± 0.1	.039	± .004	1.0	± 0.1	.039	± .004	1.0	± 0.1	.039	± .004
B3	2.00	± 0.15	.079	± .006	3.00	± 0.15	.118	± .006	11.00	± 0.15	.433	± .006
B4	5.00	+0 -0.25	.197	+0 -.010	5.00	+ 0 -0.25	.197	+ 0 -.010	5.00	+ 0 -0.25	.197	+ 0 -.010
B5	9.80	± 0.15	.386	± .006	10.80	± 0.15	.425	± .006	18.4	± 0.15	.724	± .006
B6	4.8	± 0.1	.189	± .004	4.8	± 0.1	.189	± .004	4.8	± 0.1	.189	± .004
B7	2.40	± 0.05	.094	± .002	2.40	± 0.05	.094	± .002	2.4	± 0.05	.094	± .002
B8	1.20	± 0.05	.047	± .002	1.20	± 0.05	.047	± .002	1.20	± 0.05	.047	± .002
B9	2.7	± 0.1	.106	± .004	2.7	± 0.1	.106	± .004	2.7	± 0.1	.106	± .004

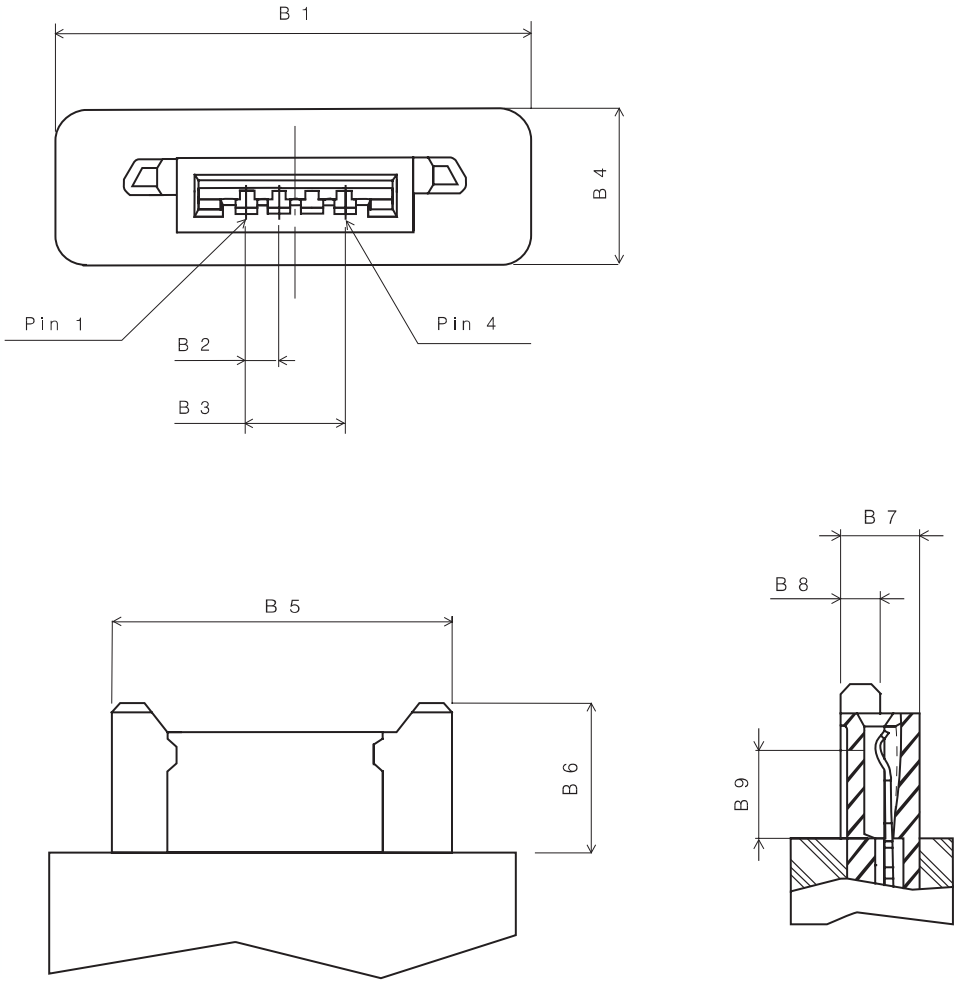


Figure 3-10: Cable Plug Connector, 4 Position

©1999PCMCIA/JEIDA

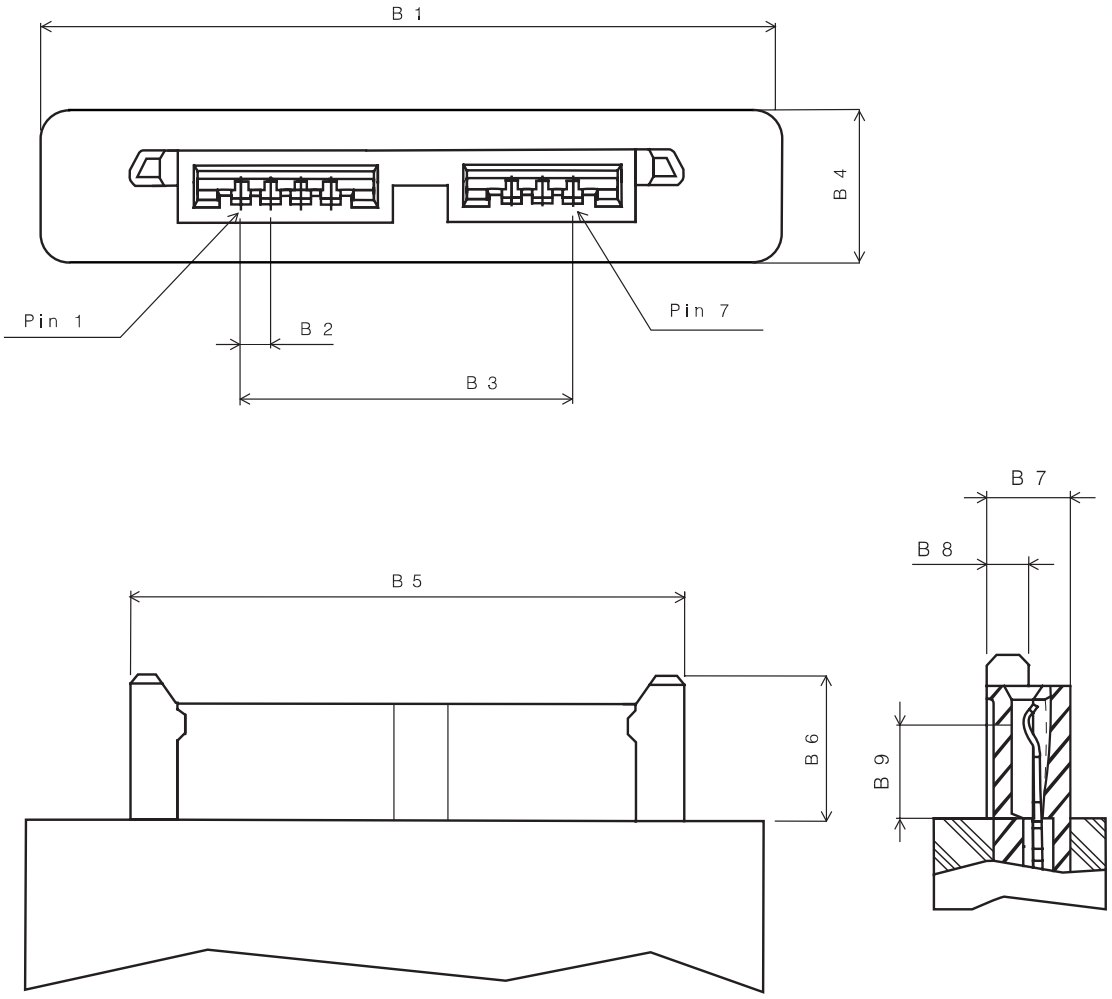


Figure 3-11: Cable Plug Connector, 7 Position

© 1999 PCMCIA/JEIDA

3.2.3.3 Materials and Finishes

3.2.3.3.1 Insulators

Materials shall be suitable for the purposes intended including the following:

Material: Shall be UL approved.

Flammability: Per UL94V-0.

PWB termination: Card Connectors shall be suitable for Surface Mounting, including (but not limited to) IR at 250 C for 10 seconds, without damage.

3.2.3.3.2 Contacts

3.2.3.3.2.1 Materials

Shall be suitable for the purposes intended.

3.2.3.3.2.2 Plating and Finish

Contact plating in engaging areas shall be nickel-palladium alloy with gold flash overplate, with suitable underplate. Plating of terminals on the card connectors shall be suitable for reflow mounting. Wire termination portions of the cable plugs shall be suitably plated using materials that are compatible with plating materials elsewhere on the contact. If lubrication is applied, it shall be per EIA CB14; lubricated contacts are required to pass the same test requirement as non-lubricated contacts.

3.2.3.4 Polarization

Alternate polarizations are not included within this specification. Connector versions similar to those per this specification that differ only by schemes of alternate polarization of keying are deemed to be in non-conformance with this specification. Such connectors are not suitable for the "open system" interface intentions of this specification. This is a different criteria than that applicable to "mis-mating" which is specified herein.

3.2.4 Test and Performance Criteria

3.2.4.1 Test Sequence

Table 3-14 defines Test Sequence. There are 17 Test Groups. Connectors per each Test Group are per the table below.

Table 3-13: Connectors Per Test Group

Group No:	1 through 11	12 and 13	14 and 15	16 and 17
Quantity Pairs	6	4	4	4
Size, Card Connector	7-position	7-position	7-position	4-position
Size, Cable Connector	7-position	4-position	3-position	4-position

©1999PCMCIA/JEIDA

Table 3-14: Test Sequence

	Test Item	Test Group																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Ref. Para No.
Test Sequence																			
1	Insulation Resistance					1, 7	1									1	1		3.2.4.4.3
2	Dielectric Withstanding Voltage					2	2									2	2		3.2.4.4.2
3	Contact Resistance (low level)		1, 5	1, 4, 6, 8	1, 4	3, 6	3, 5	1, 5	1, 3	1, 3	1	1,3, 5	1,3	1,3 ,5	1,3	3,6	3,8	1, 3	3.2.4.4.1
4	Contact Forces		2	2				2								4	4		3.2.4.3.5
5	Contact Insertion/ Withdrawal Forces		3	3				3				2		2	2		5	4	3.2.4.3.3 3.2.4.3.4
6	Contact Retention Force	1														5			3.2.4.3.6
7	Durability - Office Environment		4											4				2	3.2.4.6.1
8	Durability - Harsh Environment			5															3.2.4.6.2
9	Vibration				2														3.2.4.3.7
10	Mechanical Shock				3														3.2.4.3.8
11	Humidity (Normal Condition)			7		4													3.2.4.5.5
12	Moisture Resistance						4												3.2.4.5.2
13	Thermal Shock					5													3.2.4.5.3
14	High Temperature Life							4											3.2.4.5.4
15	Cable Strain Relief										4								3.2.4.3.11
16	Mixed Flowing Gases									2									3.2.4.5.6
17	Current Rating (capacity)										2								3.2.4.4.4
18	Inverse Mating		6								3		4				6		3.2.4.3.9
19	Mechanical Torque/ Flex											4	2				7		3.2.4.3.10
20	High Voltage Common Mode Isolation								2										3.2.4.4.6

Note: Each test group is represented by its own vertical column. In each column, the sequence of test(s) to be performed is indicated by the number 1,2,3, and so forth which indicates the order involved.

© 1999 PCMCIA/JEIDA

3.2.4.2 Standard Test Conditions

The I/O interconnect system as specified in Section 3.2.3 shall meet or exceed all reliability test requirements of this subsection. Unless otherwise specified, all tests and measurements shall be made at:

Temperature	15 °C to 35 °C
Air Pressure	86 to 106 kPa
Relative humidity	25% to 85%

If conditions must be closely controlled in order to obtain reproducible results, the parameters shall be:

Temperature	23°C ± 1C
Air Pressure	86 to 106 kPa
Relative humidity	50% ± 2%

©1999PCMCIA/JEIDA

3.2.4.3 Mechanical Performance Criteria Tests

Mechanical performance criteria are as follows, based upon testing per Table 3-14: Test Sequence.

3.2.4.3.1 Office Environment

Standard	Test
Guaranteed number of insertions/extractions = 10,000 minimum	Paragraph 3.2.4.6.1.

3.2.4.3.2 Harsh Environment

Standard	Test
Guaranteed number of insertions/extractions = 5,000 minimum	a. Paragraph 3.2.4.3.2. b. Paragraph 3.2.4.5.6.

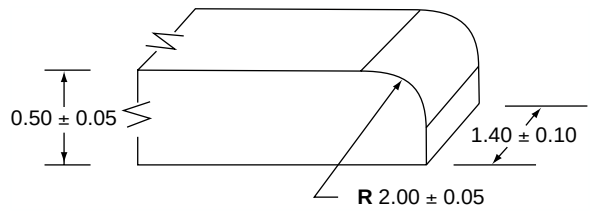
3.2.4.3.3 Total Insertion Force

Standard	Test
6.6 lb. (3.0 kg) maximum	Insertion speed: 2.54 cm/minute

3.2.4.3.4 Total Withdrawal Force

Standard	Test
0.5 kg minimum 7 position to 7 position 0.3 kg minimum all others	Extraction speed: 2.54 cm/minute

3.2.4.3.5 Single Contact Forces

Standard	Test
Using a gauge pin as shown below: 100 grams maximum insertion force. 15 grams maximum extraction force.  Tool steel gauge pin (surface finish .406μm max., chamfer all sharp edges).	Insert and extract gauge pin at a speed of 2.54 cm/minute. Perform on 2 contacts per each card and cable/plug connector, per test groups. Contacts to be non-adjacent.

3.2.4.3.6 Single Contact Retention Forces

Standard	Test
Contact shall not be displaced from insulator when a 0.5 kg minimum force is applied to the contact in the axis of retention.	Apply a minimum force of 0.5 kg to the contact in both directions along the axis of retention while holding the insulator rigid

© 1999 PCMCIA/JEIDA

3.2.4.3.7 Vibration

Standard	Test
a. No mechanical defects shall occur on the parts. b. No current interruption greater than 100 ns. c. Contact resistance per 3.2.4.4.1	MIL-STD-202F, Method 204D, Test condition B (15G peak), 10 Hz to 2000 Hz.

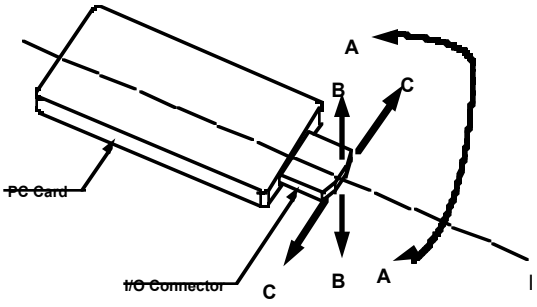
3.2.4.3.8 Shock

Standard	Test
a. No mechanical damage shall occur on the parts. b. No current interruption greater than 100 ns.	MIL-STD-202F, Method 213B, Acceleration 50G, Standard holding time 6 msec., Half-sine wave.

3.2.4.3.9 Inverse Mating

Standard	Test
Connector shells (polarization) must remain intact and functional to this requirement after durability testing. 7 position mated to 7 position @ 6 kg 3 or 4 position mated to 7 position @ 3 kg 4 position mated to 4 position @ 3 kg Note: Connectors shall pass if 3 or 4 position shells yield when mated to 7 position, provided that 7 position connector is not affected.	Connector system must not mate when cable plug connector is mis-mated upside down to card I/O connector, for 1 minute. Repeat 5 times.

3.2.4.3.10 Connector Plug Torque and Flex

Standard	Test
a. No mechanical damage shall occur on the parts. b. No current interruption greater than 100 nsec.  <u>Torque Requirements:</u> Mated 7 position pair @ 0.23 N. Mated 3 or 4 position to 7 position or mated 4 position pair @ 0.11 N	1. Clamp PC Card at I/O connector end, within 5 cm of the card edge. I/O connector pair shall be mated per Table 3. 2. Apply clockwise torque of 0.23 N to the cabled end of the connector, hold at extreme for 5 minutes. Repeat test in counter-clockwise direction. Repeat 5 times in each direction. 3. Apply upward torque of 0.23 N to the cable end of the connector. Hold for 5 minutes. Repeat in the downward, left and right directions. Repeat 5 times in all 4 directions. 4. Apply upward torque of 0.23 N to the cable end of the connector. Hold for 3 seconds. Repeat in the downward, left and right directions. Repeat 1,000 times in all 4 directions.

©1999PCMCIA/JEIDA

3.2.4.3.11 Strain Relief

Standard	Test
The strain relief shall be capable of retaining the attached cable with no separation or damage to conductor(s), shielding (if applicable), or insulation. Connector shall be terminated with strain relief in manner used for standard production	Clamp connector housing. Apply force to cable outward from connector housing. Force shall be 6.4 kg for 5 minutes.

3.2.4.4 Electrical Performance Criteria

Electrical criteria, for testing per Table 3-14: Test Sequence, are as follows:

3.2.4.4.1 Contact Resistance (low level)

Standard	Test
a. Initial...40 mΩ maximum. b. After Test...45 mΩ maximum. c. Contact resistance increase shall not exceed 20 mΩ total.	MIL-STD-1344A, Method 3002.1, Open voltage 20 mV. Test current: 1 mA. a. Measure and record the initial resistance (R_1) of the connector system (from attachment of male connector to the PCB, to the wire of the cable plug connector). $R_1 \leq 40$ mΩ b. Measure and record resistance (R_f) of the connector system after test of the connector system. Resistance value after test: $R_f \leq 45$ mΩ

3.2.4.4.2 Dielectric Withstanding Voltage

Standard	Test
a. No shorting, breakdown, flash-over or other damage when 600 Vrms ac is applied for 1 minute. b. Current leakage 1mA maximum.	MIL-STD-202F, Method 301, measured between adjacent contacts.

3.2.4.4.3 Insulation Resistance

Standard	Test
a. Initial.....1,000 MΩ minimum. b. After test...100 MΩ minimum.	MIL-STD-202F, Method 302, measured within 1 minute after applying 250 Vdc.

3.2.4.4.4 Current Capacity

Standard	Test
0.5 A per contact.	20 °C maximum Temperature rise over ambient at rated current.

© 1999 PCMCIA/JEIDA

3.2.4.4.5 Insulation Material

Standard
UL94V-O rated material in thickness used.

3.2.4.4.6 High Voltage Common Mode Isolation

Standard	Test
No arcing, 10 mA AC rms maximum current to 3750 V rms.	IEC 950, Section 5.3.2. except test to 4125 V rms, minimum. (sea level conditions).

3.2.4.5 Environmental Performance Criteria

3.2.4.5.1 Environmental Conditions

Operating Temperature Range	-20^{O}C to +60^{O}C
Storage Temperature Range	-20^{O}C to +70^{O}C
Relative Humidity	95% maximum (non-condensing)

3.2.4.5.2 Moisture Resistance

Standard	Test
Contact Resistance: 3.2.4.4.1. Insulation Resistance: 3.2.4.4.3.	MIL-STD-202F, Method 106E (excluding vibration); 10 cycles (1 cycle = 24 hours) with connector system mated.

3.2.4.5.3 Thermal Shock

Standard	Test
No physical damage shall occur during testing. Contact resistance: 3.2.4.4.1. Insulation resistance: 3.2.4.4.3.	MIL-STD-202F, Method 107G, Test condition A, -55^{O}C to +70^{O}C, 5 cycles (1 cycle = 1 hour) with connector mated.

3.2.4.5.4 Durability (High Temperature)

Standard	Test
Contact resistance: 3.2.4.4.1.	MIL-STD-202F, Method 108A, Test condition B, +70^{O}C, 250 hours minimum, with connector mated.

©1999PCMCIA/JEIDA

3.2.4.5.5 Humidity (Normal Condition)

Standard	Test
Contact resistance: 3.2.4.4.1. Insulation resistance: 3.2.4.4.3.	MIL-STD-202F, Method 103B, Test condition B, +40°C, 90% to 95 % RH, with connector mated.

3.2.4.5.6 Mixed Flowing Gas

Standard	Test
a. No visible corrosion under 3X magnification. b. Low level contact resistance per 3.2.4.4.1. except maximum increase to be 20 mΩ above initial measurements.	EIA-364-65 (TP-65), Class II environment for 48 hours after stabilization. Performed with 50% of test group connectors mated; rest unmated. Measurements made on 100% of contacts (all connectors per group).

3.2.4.6 Durability Mating Requirements

The I/O interconnect system as specified in Section 3.2.3 shall meet or exceed all durability requirements of the subsection.

Test conditions for the mate/unmate cycles are:

Cycle rate	400-600 cycles per hour
Temperature	15 °C to 35 °C
Air Pressure	86 to 106 kPa
Relative Humidity (RH)	25% to 85%

3.2.4.6.1 Office Environment

The office environment as defined in EIA-364 B as Class 1.1, year round air conditioning (nonfiltered) with humidity control.

Test Sequence:

Contact resistance per 3.2.4.4.1.
Mate and unmate the connector for a total of 10,000 cycles.
Contact resistance per 3.2.4.4.1.

© 1999 PCMCIA/JEIDA

3.2.4.6.2 Harsh Environment

The harsh environment as defined in EIA-364B as Class 1.3, is no air conditioning, no humidity control with normal heating and ventilation, plus consideration for industrial environment conditions.

Test Sequence:

Contact resistance per 3.2.4.4.1.	
Mate and unmate the connector for a total of 1,000 cycles.	Total cycles = 1,000 cycles
Humidity per 3.2.4.5.5	
Contact resistance per 3.2.4.4.1.	
Mate and unmate the connector for a total of 1,000 cycles.	Total cycles = 2,000 cycles
Humidity per 3.2.4.5.5.	
Contact resistance per 3.2.4.4.1.	
Mate and unmate the connector for a total of 3,000 cycles.	Total cycles = 5,000 cycles
Humidity per 3.2.4.5.5.	
Contact resistance per 3.2.4.4.1.	

Separate Test Sequence:

Contact resistance per 3.2.4.4.1.
Mixed Flowing Gas per 3.2.4.5.6.
Contact resistance per 3.2.4.4.1.

©1999PCMCIA/JEIDA

3.2.5 Contact Position Assignment

Contact position assignments shall conform to paragraph 3.2.5.1. Card Jack and Plug connectors shall be marked to identify contact locations as follows: for 7 position PC Cards: 1, 4, 5 and 7; 4 position PC Cards: 1 and 4: 3 position PC Cards (plug only): 5 and 7. Refer to Section 3.2.7 for application specific Signal-Contact interconnection assignments.

3.2.5.1 Contact Position Marking

Contact position shall be clearly marked as to be visible to the user.

3.2.5.2 Cable Interconnections

Four basic cable types may be configured based upon the three plug connector types. All physical wiring interconnections are to be point-to-point. In all cases, the "contact position assignments" shall be per the applicable "basic cables" shown below. Information regarding application parameters are included in Section 3.2.7.1. Cable types shall be as defined in the following table.

Table 3-15: Cable Type, Point-to-Point Wiring Physical Definition

Cable Type	Receptacle (Card side)	Plug (Cable side)
TYPE A 7 pos. plug to 7 pos. card	1 2 3 4 5 6 7	1 2 3 4 5 6 7
TYPE B 4 pos. plug to 7 pos. card	1 2 3 4 5 6 7	1 2 3 4 NC NC NC
TYPE B.2 4 pos. plug to 4 pos. card	1 2 3 4	1 2 3 4
TYPE C 3 pos. plug to 7 pos. card	1 2 3 4 5 6 7	NC NC NC NC 5 6 7

Note: "**NC**" indicates "No Connection."

© 1999 PCMCIA/JEIDA

3.2.6 Intermatability

Intermateability between manufacturers shall be confirmed by the use of 50% of connectors from other manufacturer(s) during testing of Group 2, 5, 7, 8, 10, 14, and 16. This shall involve both card and cable connectors from all manufacturers. This may be done as part of the manufacturer's standard qualification tests.

3.2.6.1 Intermateability Assurance

It shall be the responsibility of each manufacturer to confirm intermateability. By reference to this document, PCMCIA is presenting an "open-system" interface to permit interchangeability of components between suppliers. Due to the options that may be exercised by various manufacturers, PCMCIA does not define this interface beyond the limited criteria set forth by this document.

©1999PCMCIA/JEIDA

3.2.7 PC Card Modem 4 Pin I/O for PSTN Connectivity

This recommendation defines the pinout of a 4 pin I/O connector for use in PC Card modems equipped with internal DAAs and connecting directly to the Public Switched Telephone Network (PSTN) and which are intended for approval in the United States and Canada. It assumes the use of the 4 pin I/O connector defined in Section 3.2.3 of this Standard. PC Card Modems equipped with this connector require only a cable to attach to the customer premise modular RJ11 Telephone Jack. It defines the use of a four position connector on the PC Card modem and a Telco cable which connects pins 2 and 3 of the 4 pin mating plug to pins 3 and 4 of a modular RJ11 plug. Simultaneous connection of the modem to the teleco and to a local handset can be accomplished by providing a cable assembly which connects pins 2 and 3 of a 4 pin mating plug to pins 3 and 4 of a modular RJ11 plug and connects pins 1 and 4 of the same 4 pin mating plug to pins 3 and 4 of a second modular RJ11 plug. By following this recommendation, it is possible to create unique standard cable assemblies that provide one or both of the functions listed above.

3.2.7.1 Pinout Configuration for 4 Pin PC Card Modem I/O Connector

The pinout defined for Modem connector is shown in the table below.

Table 3-16: 4 Pin PC Card Modem I/O Connector Pinouts

Pin	Signal	I/O	Function	+/-
1	Tip 1	Analog	The tip signal to the teleset	Balanced
2	T	Analog	The tip signal normally connecting to pin 3 of the RJ11 Plug	Balanced
3	R	Analog	The ring signal normally connecting to pin 4 of the RJ11 Plug	Balanced
4	Ring 1	Analog	The ring signal to the teleset	Balanced

3.2.8 PC Card Modem 7 Pin I/O with Audio Interface

This recommendation defines the pinout of a 7 pin I/O connector for use in PC Card modems (as defined in Section 3.7 of this Standard) and includes the definition of an interface appropriate for use with an audio device. It assumes the use of the 7 pin I/O connector defined in Section 3.3 of this Standard. PC Card Modems equipped with this connector require only a cable to attach to the customer premise modular RJ11 Telephone Jack for PSTN connectivity in the manner defined in Section 3.7 as applied to the 4 pin section of the 7 pin connector. The 3 pin section of the connector is defined for use with commonly available audio headsets, but could just as well be interfaced to a non-headset style appliance duplicating the expected audio input and output functions.

This audio interface is defined to support modem-based voice and business audio applications including hands-free phone, telephone answering machine (TAM), simultaneous or alternating voice/data (SAVD), voice annotation, and presentation audio applications. Although this interface is intended primarily for audio headset applications, it is also applicable as an interface to an externally powered speaker/mic accessory and thus also useable for group applications including speakerphone and business presentations.

3.2.8.1 Pinout Configuration for 7 Pin PC Card Modem I/O Connector

The pinout defined for Modem with Audio connector is shown in the table below. This pinout supports having an audio interface in addition to the modem's PSTN interface. Regarding the

audio interface portion of the pinout, refer to Section ***3.2.8.2 Electrical Characteristics for the Audio Interface*** for a definition of the electrical characteristics.

Table 3-17: 7 Pin PC Card Modem I/O Connector Pinouts

Pin	Signal	I/O	Function	+/-
1	Tip 1	Analog	The tip signal to the teleset	Balanced
2	T	Analog	The tip signal normally connecting to pin 3 of the RJ11 Plug	Balanced
3	R	Analog	The ring signal normally connecting to pin 4 of the RJ11 Plug	Balanced
4	Ring 1	Analog	The ring signal to the teleset	Balanced
5	Audio Out	Analog	The audio signal to be sent to a speaker	+
6	Return	Analog	Return path for both audio in and out signals	Ground
7	Audio In	Analog	The audio signal to be supplied by a microphone	+

3.2.8.2 Electrical Characteristics for the Audio Interface

The following electrical characteristics are defined for each of the audio signals.

Audio Output (speaker): Series Resistance = 10 ohms (minimum, for current limiting); capable of driving up to 1.0 volt peak-to-peak into a speaker load of 100 ohms.

Audio Input (microphone): The microphone is anticipated to be electret, therefore the audio input will require a bias circuit, an example of which is shown in ***Figure 3-12: Microphone Input with Bias Supply***. The microphone input should be capable of accepting voltage peak-to-peak inputs ranging from 10 mv to 1000 mv. The recommended minimum input frequency range of the preamp is 300 Hz to 3 kHz. To reduce noise, it is strongly recommended that the bias voltage be decoupled as shown in the example figure.

As microphone input levels will vary with different headsets, it is recommended that the preamp section for the microphone input have a method for adjusting the gain to optimize sound levels over the voltage peak-to-peak range indicated above.

Audio Return: Tied directly to the PC Card's signal ground.

 ©1999PCMCIA/JEIDA

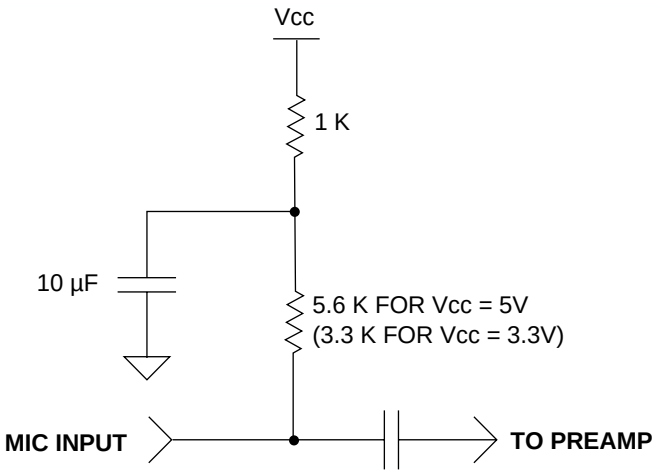


Figure 3-12: Microphone Input with Bias Supply

© 1999 PCMCIA/JEIDA

3.3 Guideline for Maximum Dimensions for I/O Connectors

3.3.1 Background

Although many kinds of I/O connectors are available on the market, there are no limitations on I/O connector dimensions, making PC Card slot designs difficult. This guideline describes the recommendation for Maximum dimensions for an I/O connector. Taking into consideration the maximum dimensions for an I/O connector helps the design of both PC Card Slot bezels and new I/O connectors.

3.3.2 Guideline

The following dimensions, A to E for I/O connector are shown in ***Figure 3-13: Maximum Dimensions for I/O Connector*** below.

Note: This Guideline takes into consideration single PC Card Slots only. For multiple PC Card Slots, you need to consider appropriate numbers for B1, B2, C and D.

A:	PC Card I/O connector length within PC system enclosure area.
B1, B2:	PC Card I/O connector height in enclosure area referenced to centerline of PC Card.
C, D :	PC Card I/O connector cabled plug height external to system enclosure area referenced to centerline of PC Card.
E :	Width of PC Card I/O connector within system enclosure area A.

©1999PCMCIA/JEIDA

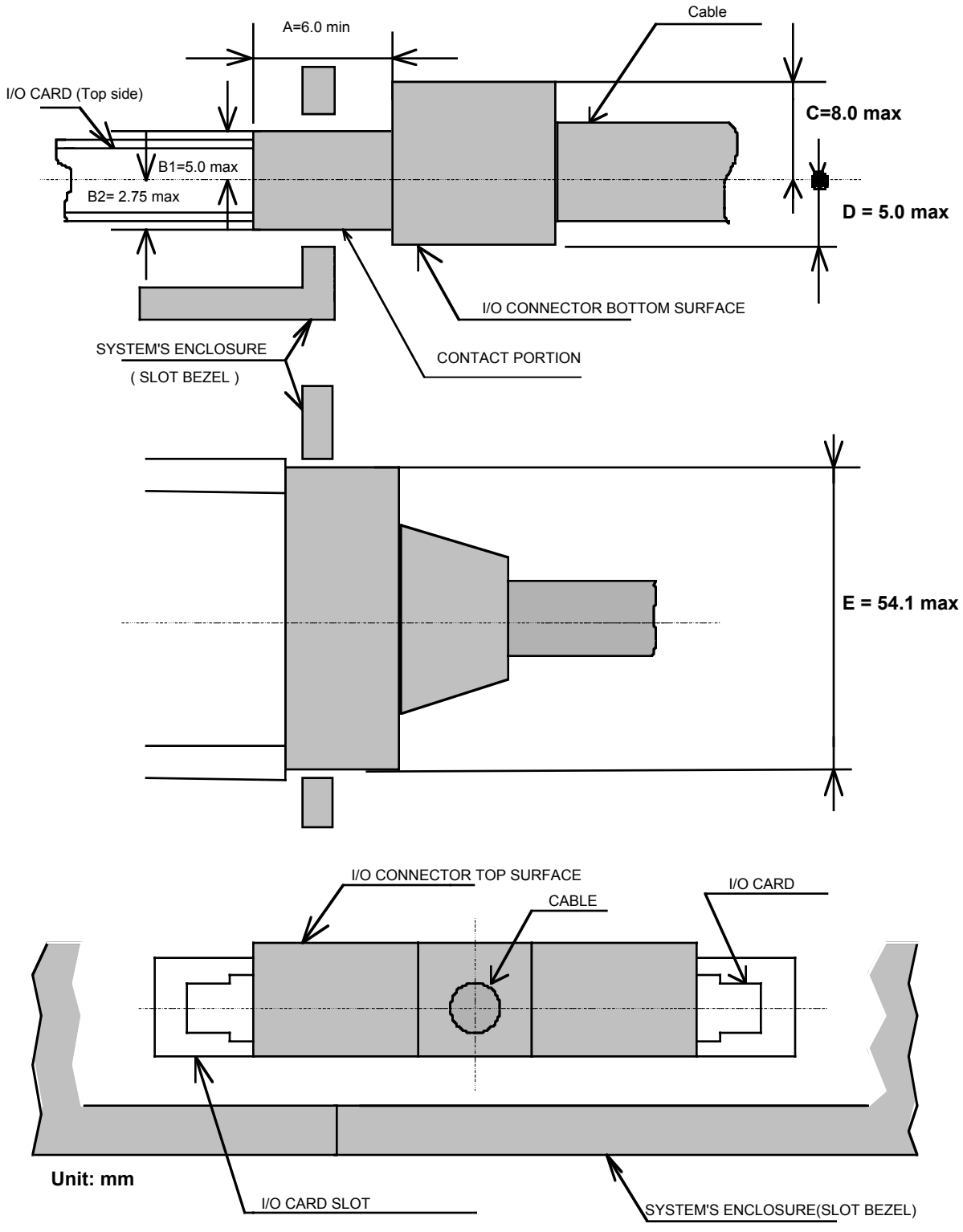


Figure 3-13: Maximum Dimensions for I/O Connector

Note: This figure defines single slot only.

© 1999 PCMCIA/JEIDA

3.4 Extended PC Card Guideline

3.4.1 Summary

The Type I and II Extended PC Card outlines are identical to the Type I and II PC Card outlines except for the extended portion. The extended portion extends 10 mm past the standard PC Card length of 85.6 mm before the height may be increased in the bubble area. It should be noted that in the extended bubble portion the thickness from the centerline of the connector to the bottom of the PC Card is the same thickness for the entire length. The centerline of the connector to the top of the bubble is 8.0 mm recommended. This will allow a thicker bubble.

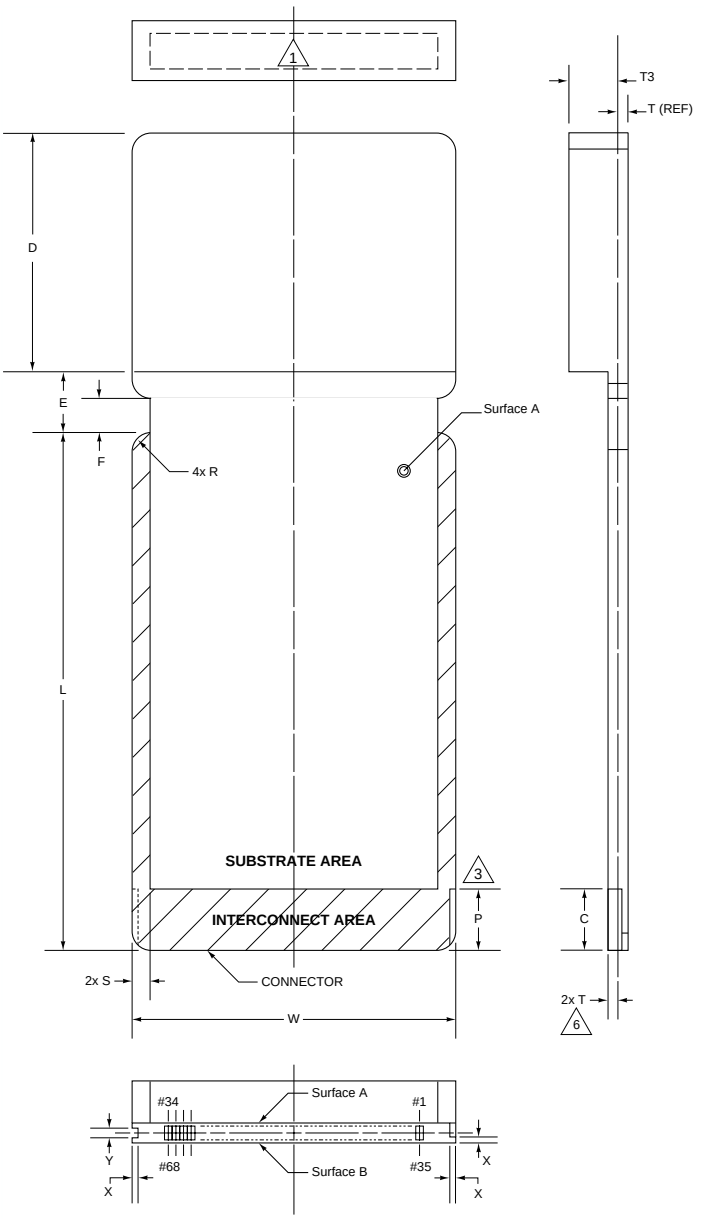
3.4.2 Background

The Type I and II Extended PC Card outlines were primarily specified so that designers and manufacturers of I/O PC Cards may encase their electrical and magnetic isolation devices within the shielded PC Card enclosure. It was noted during the discussion before passage of the Type I and II Extended PC Card Outline that some connectors (RJ-11 and RJ-45) will not fit within the recommended thickness. Therefore, PCMCIA's Card Physical Committee specified the thickness such that the centerline of the connector to the bottom of the PC Card is 2.5 mm max. The height from the socket centerline to the tip of the bubble is 8.0 mm recommended. This recommended height will allow the designer to increase the total thickness of the bubble in order to accommodate the RJ-11 and RJ-45 connectors.

3.4.3 Guideline

The following figures define the recommended outline for Type I or Type II Extended PC Cards.

©1999PCMCIA/JEIDA



C MIN	L ± 0.20	P MIN 3	T 4 6	S MIN	W ± 0.10	X ± 0.05
10.0	85.60	10.0	1.65	3.0	54.0	1.00
Y ± 0.05	D MAX	E ± 0.20	F MIN	R MAX	T3 5	
1.60	40.0	10.0	5.0	3.0	8.0	

1 RECOMMENDED I/O CONNECTORLOCATION.

2. THE PC CARD SHALL BE OPAQUE (NON SEE THRU)

3 POLARIZATION KEY LENGTH.

4 INTERCONNECT AREA TOLERANCE = ± 0.05
SUBSTRATE AREA TOLERANCE = ± 0.10

5 RECOMMENDED MAX DIMENSION.

6 FOR CARDBUS PC CARDS, DIMSENTION T IS INCREASED BY 0.50 ± 0.05 OVER DIMPLES
(REFER TO CARDBUS PC CARD RECOMMENDED CONNECTOR GOUNDING INTERFACE DIMENSIONS
FIGURE IN THE PHYSICAL SPECIFICATION)

Figure 3-14: Type I Extended PC Card Package Dimensions

© 1999 PCMCIA/JEIDA

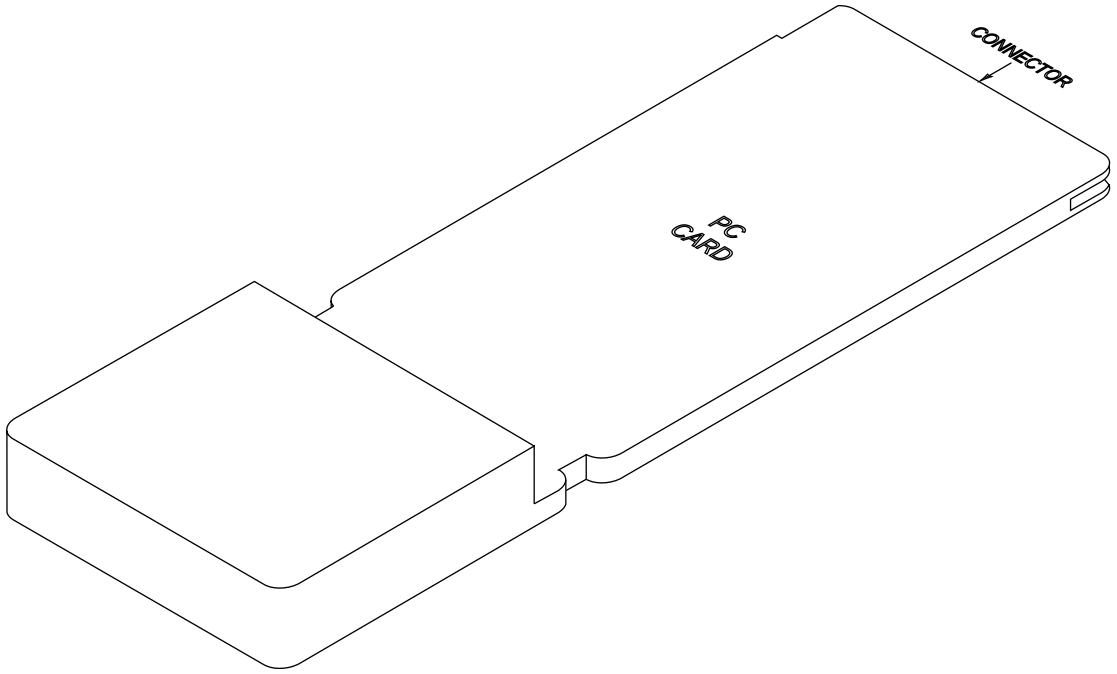
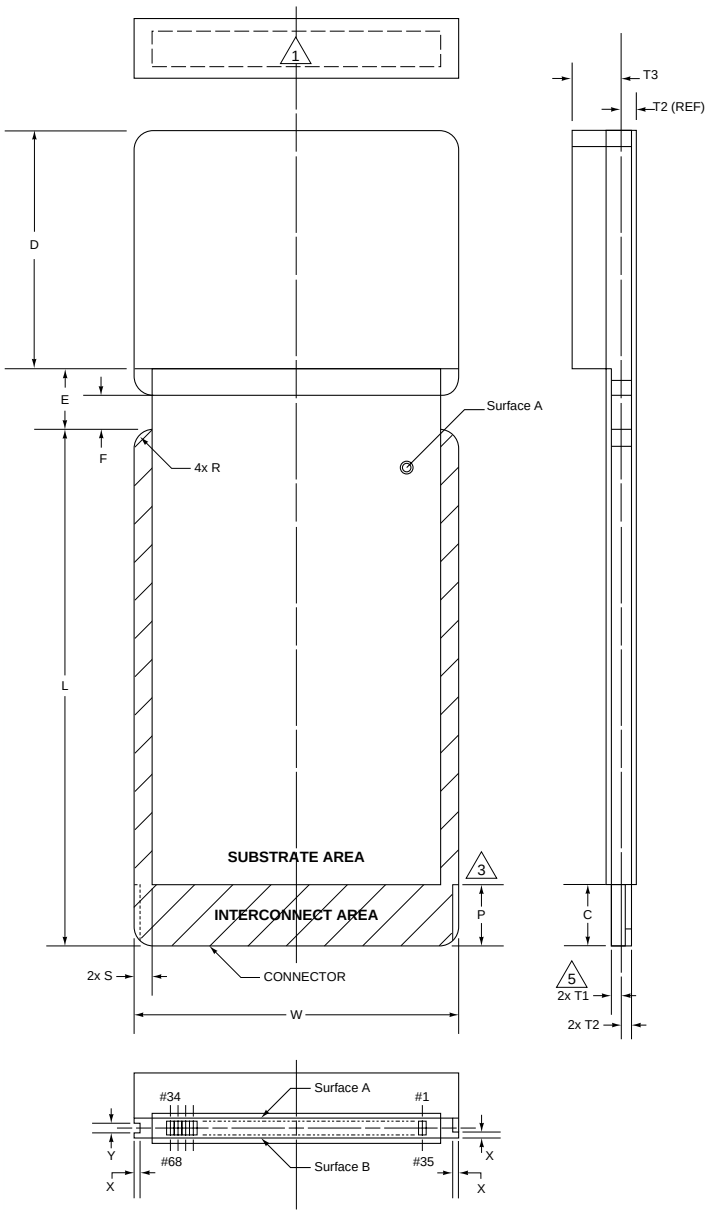


Figure 3-15: Type I Extended (3-D)

©1999PCMCIA/JEIDA



C MIN	L ± 0.20	P MIN 3	T1 ± 0.05 4	T2 MAX	S MIN	W ± 0.10
10.0	85.60	10.0	1.65	2.50	3.0	54.0
X ± 0.05	Y ± 0.05	D MAX	E ± 0.20	F MIN	R MAX	T3 4
1.00	1.60	40.0	10.0	5.0	3.0	8.0

1. RECOMMENDED I/O CONNECTOR LOCATION.
2. THE PC CARD SHALL BE OPAQUE (NON SEE THRU)
3. POLARIZATION KEY LENGTH.
4. INTERCONNECT AREA TOLERANCE = ± 0.05
   SUBSTRATE AREA TOLERANCE = ± 0.10
5. FOR CARDBUS PC CARDS, DIMENSION T1 IS INCREASED BY 0.50 ± 0.05 OVER DIMPLES
   (REFER TO CARDBUS PC CARD RECOMMENDED CONNECTOR GROUNDING INTERFACE DIMENSIONS
   FIGURE IN THE PC CARD STANDARD PHYSICAL SPECIFICATION)

Figure 3-16: Type II Extended PC Card Package Dimensions

© 1999 PCMCIA/JEIDA

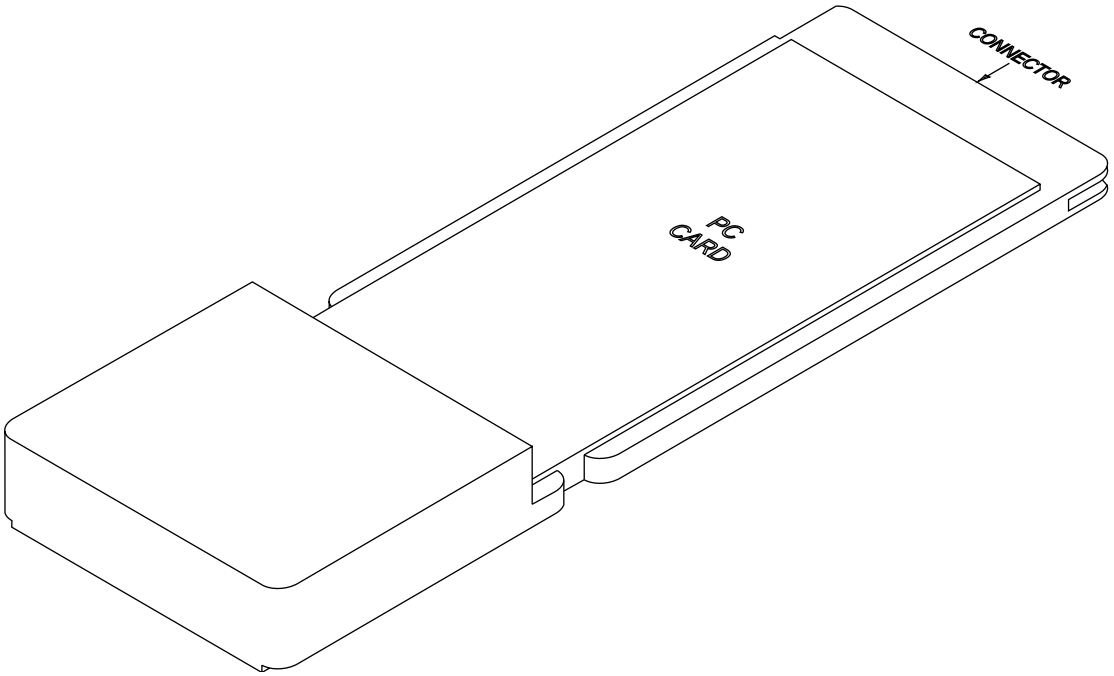


Figure 3-17: Type II Extended (3-D)

©1999PCMCIA/JEIDA

4. SOFTWARE GUIDELINES

4.1 Enabler Capabilities and Behavior

4.1.1 Summary

This guideline recommends the capabilities that all enablers should provide and proscribes certain behavior. Following this guideline minimizes colliding accesses between the enabler and other clients and between independent enablers thus guaranteeing a minimum level of functionality to users.

4.1.2 Background

The purpose of an enabler is to allow a single piece of software to configure and unconfigure a PC Card to prevent the redundancy of every PC Card requiring separate and unique configuration software.

Enablers recognize PC Cards in one of two ways. First, they may recognize a card specifically from information in the Card Information Structure (CIS) that is unique to that card. For example, some enablers use combinations of the manufacturer string and product string from the VERS_1 tuple and the Manufacturer ID and Function ID tuples. Second, enablers may recognize a PC Card solely from its Function ID tuple. For most Data/FAX modems, a Function ID that indicates the PC Card has a serial interface causes the card to be configured as the next available COM port.

Once a card is configured, it may be used by software that is or is not aware that the functionality is located on a PC Card. If the software is PC Card-aware, it may piggy-back on the efforts of the enabler and use the card's functionality until the card is removed and the enabler releases the system resources allocated to the card.

If the software is not PC Card-aware, it must locate the card's resources where the enabler mapped them into the host system's address space. For most cards this means assuming the PC Card is placed at the same locations typically used by standard ISA peripherals that perform the same functions.

Some cards offer functionality that use configuration files to describe where they are located in system memory. As an example, network adapters have a NET.CFG or PROTOCOL.INI file describing the IRQ level, I/O address range and memory range used to access the adapter. In this case, the enabler must place the PC Card's resources as indicated in the configuration file.

An enabler which follows this guideline should be able to properly configure any card which follows the CIS requirements for its card type and contains logically correct and consistent information within the tuples.

Note that although an Enabler API is not prohibited, such an API is not described by this guideline.

4.1.3 Guideline

It is strongly recommended that PC Card system software implementations provide an enabler. If such an enabler is present in a system, it should be tested as part of that system's integration test.

Enablers developed for the PC Card environment should obey the following guidelines:

1. Enablers should support 16-bit PC Cards as well as CardBus PC Cards. The use of a single enabler reduces usage of system resources and simplifies software interactions. This goal cannot be achieved without building support for both types of cards in the enabler. If it is necessary for there to be more than one enabler resident in the system in order for the system to support both types of cards, then rule #3 should be observed.
2. Enablers should be able to configure cards which meet the CIS requirements outlined in the ***Metaformat Specification***. This ensures that cards designed to the minimum requirements can be configured by all enablers designed to support that class of card.
3. Enablers should not interfere with the registration and configuration of Card Services clients which do not use the enabler. It is recommended that enablers honor exclusive requests for a function. Therefore, there should be a mechanism for *not* generically enabling a card even though it could be enabled. Enablers should allow users to designate which cards or which class of cards it will not configure.
4. Enablers should never cause any client to lose resources unwillingly because resource changes cannot be handled gracefully (i.e., transparent to the user) without the involvement of the client.
5. If the system allows the user to query, then enablers should be able to inform users about resource and logical device configuration. This provides the ability to notify the user of card configuration status.
6. All system software must be aware of its impact on the system resources. For example, in an x86 type of system, enablers should make as much use as possible of memory above 1 MByte (high and/or extended memory) to minimize the impact on the first 640 KBytes of system memory (low memory).
7. Card Services is the owner of all PC Card resource management for clients including the enabler. If an enabler uses a private API for assigning resources, it should inform Card Services of the outcome of the arbitration.
8. Enablers which don't enable certain classes of cards must not configure or modify the configuration of that class of cards. For example, an enabler which only manages memory cards must not configure or modify the configuration of I/O cards.
9. In CardBus PC Cards, memory windows are a fixed size so no standard means exists to reassign windows. Enablers which also act as 16-bit PC Card memory clients must be able to reassign, reuse, and resize the same host memory space to accommodate additional memory cards so that each memory client can use (allocate and deallocate) the same memory address space. For performance reasons, memory cards will want as much space as is available. However, the appearance of new clients causes a need to dynamically resize these windows.

©1999PCMCIA/JEIDA

4.2 Card–Application Interaction

4.2.1 Summary

This guideline minimally outlines how PC Card-aware and PC Card-unaware applications should interact with a PC Card.

4.2.2 Background

The following terms are used in this Guideline:

Application Function Test	are the tests that an application tester would normally use to confirm that an application basically works as designed. They are not comprehensive tests and do not necessarily address boundary and error conditions.
Enabler	is also referred to as a "generic enabler." This is a Card Services client which is capable of configuring a variety of cards. It may or may not have other capabilities such as the following: • providing an alternate API to Card Services • providing the user information about the installed cards This type of enabler is not custom designed for configuring specific cards, but, using the CIS and the Card Services interface, it can configure many different cards, or classes/types of cards.
PC Card-Aware Application	is an application which has been modified to take into account the PC Card bus's more dynamic environment. This would include having either the application or a driver the application uses becoming a Card Services client.
PC Card-Unaware Application	is an application which has not been modified as above.

In any given PC Card system, there can be both PC Card-aware and PC Card-unaware applications. Both of these types of applications may wish to access the PC Card, and could react very differently to changes in the card status. This guideline attempts to define a minimum standard of good behavior in several simple, common situations.

© 1999 PCMCIA/JEIDA

4.2.3 Guideline

4.2.3.1 PC Card-Unaware Applications

For PC Card-unaware applications, the PC Card will have to be configured by some PC Card-aware software, usually an enabler. After the card is configured, the application will behave towards the card as if it were in a static bus environment. Following are the guidelines for the just described situation:

1. The PC Card should be able to be used with an appropriate well-known/well-behaved PC Card-unaware application (e.g., communications software for a modem card). The application should be able to pass its basic functional test while using the card as long as the card is neither inserted nor removed during application operation.
2. It is acceptable that the user may have to intervene to appropriately configure the application to access the PC Card (e.g., tell it which COM port the enabler has chosen for the card).
3. The user may have a need to know about how a particular card has been configured, as in the COM port example above. The enabler that configured that card must be able to provide that information on request.

Furthermore:

1. once the card and the application have been configured
2. and provided that the card is not removed during execution

There should be no discernible difference between a PC Card-unaware application on a PC Card system using a PC Card and a PC Card-unaware application on a non-PC Card system using a static card.

4.2.3.2 PC Card-Aware Applications

The PC Card-aware application has an interface to Card Services via one of the following pieces of software, which will configure any PC Card it uses:

- an enabler
- a device driver
- the application itself

Following are the guidelines for the just described situation:

1. The card should be able to be used with an appropriate well-known/well-behaved PC Card-aware application. The application should be able to pass its basic functional test while using that card. The application functional test should still pass if the card is inserted or removed during application operation.
2. A PC Card-aware application exhibits its normal behavior while a PC Card it is using or wants to use is removed or inserted. In general, it will treat these events in the same way that it would for floppy removal and insertion. In some cases, this treatment may not be possible; for example, when a network connection or some similar resource is lost. In this instance, however, the application should not crash, and then should

   a) inform the user of the event

   b) allow the user to recover gracefully

 ©1999PCMCIA/JEIDA

c) not interfere with the operation of other applications

3. The PC Card should be able to be installed and configured by its designated configuring software with minimal to no user intervention. The card could be configured by an enabler or by software specifically designed for the card such as a device driver. It is possible that a card could be configured by more than one piece of software.
4. Unless exclusive access has been requested, PC Card-aware applications should be able to share PC Cards. If the application can not share the card, then it should request exclusive access to the card.
5. Some PC Cards may be able to be configured by more than one Card Services client, including one or more enablers. For such cards, the user must be allowed to designate which software will configure and/or manage that card. This choice should be made when the card is inserted the first time and need not be repeated on successive insertions.
6. As with PC Card-unaware applications the user may have a need to know about how a particular card has been configured. The software that configured that card must be able to provide that information on request. This information may be provided in a variety of ways. The full extent of the information should not be announced every time the card is inserted, in such a way that the operation of the application that the user is executing is continually interrupted. Depending on implementation, notification of logical device allocation may be an exception to this general rule of application non-interruption, as it may be needed immediately by the user.
7. Many PC Card-aware applications and/or their drivers may only work with specific versions of PC Card hardware and software, e.g., a Card Services conforming to a given level of the specification. If a Card Services client is incompatible with any resident hardware or software, it must immediately report that finding to the user and gracefully stop its continuing execution. Trying to continue execution with incompatible components is ultimately more distressing to the user since the cause of the failure, when it happens, will not be as obvious as it would have been initially.

© 1999 PCMCIA/JEIDA

4.3 CardBus Operational Scenarios

4.3.1 Summary

The client to Card Services interface for CardBus PC Cards is very similar to the interface for 16-bit PC Cards. The processing done by the PC Card software stack is different for CardBus PC Cards in order to maintain that common interface. The intent of this guideline is to give an example of how such processing may be done.

4.3.2 Background

CardBus sockets support both 16-bit PC Cards and CardBus PC Cards. Clients for these cards both use the same Card Services interface. However, because the on-card programming model is so different for CardBus PC Cards, the PC Card software stack needs to internally process CardBus configuration requests differently.

There is no one way to implement CardBus processing. For example, a designer might choose to apply the minimum configuration power upon card insertion or only after a request is made to read the CIS. This guideline is meant to be one example of how an implementation might be done.

4.3.3 Guideline

The following scenario describes the recognition and configuration of a CardBus PC Card. Via the CardBus Socket Services, Card Services knows that a CardBus PC Card has arrived. At this time, it performs the following actions:

1. Card Services notifies its registered clients that a card has arrived. This notification is sent for each function on the card, so that each function looks like a separate card to a client.
2. The client requests tuples using the socket and function number it was sent in its insertion notification.

   a) The client requests the first tuple of the function via Card Services.

   CardBus clients may be drivers, applications, generic enablers, etc. just like 16-bit PC Card clients. Additionally, because some of the cards implemented in CardBus may share silicon with PCI cards, CardBus drivers may also be modified versions of PCI drivers. See the guideline *CardBus/PCI Common Silicon Requirements* for some considerations when developing this type of software.

   b) Card Services applies power to the socket in order to read the CIS. For CardBus, all cards are required to be able to support CIS reads at 70 mA. At this point, if the adapter has not been previously configured, Card Services will configure the adapter before proceeding.

   c) Card Services reads the CIS pointer from configuration space to find the beginning of the first chain of the CIS. The CIS pointer indicates the card space in which the chain resides:

      i. configuration space

      ii. one of the six (6) possible memory spaces

      iii. the expansion ROM

      and an offset into the space where the chain actually starts

 ©1999PCMCIA/JEIDA

d) If the indicated chain starts anywhere but configuration space, Card Services maps the indicated card space into host system address space by writing the appropriate Base Address Register. Note that this is a temporary mapping only. Mappings become permanent only when the client requests them as part of the configuration.

e) If the client has requested to see all link-related tuples, then the first tuple will always be a CISTPL_LINKTARGET tuple.

f) The client requests the next tuple.

g) The client walks through the function's CIS just as a 16-bit PC Card client would. For any tuples in non-configuration space, the client may read them directly. However, it is strongly recommended that the client use the Card Services interface and not read the tuples directly.

Note that in order for a client to read the tuples directly it needs to be able to perform the following tasks which Card Services would normally perform for it:

i. temporarily mapping in the card space in order to read the tuples

ii. interpret the CISTPL_LONGLINK_CB tuple to find additional chains in the card spaces.

3. If the client recognizes the card as one it is interested in, the client then uses GetConfigurationInfo for that function to see if it has been configured.
4. The client requests resources for the function from Card Services.
5. If the requests are successful, the client obtains the configuration for the function.

   For all address space requests (e.g., I/O, memory, expansion ROM), Card Services writes the appropriate value (a host address) into a Base Address Register. It does not request windows from Socket Services.
6. After the function is configured the client may now access the function.

In this scenario, clients may operate, as far as configuration is concerned, completely as 16-bit PC Card clients. This includes using a generic enabler if the client does not wish to do the configuration requests itself.

Unchanged 16-bit PC Card software continues to work in the expected manner when a 16-bit PC Card is inserted in a CardBus PC Card socket. In addition to sockets with both CardBus and 16-bit PC Card capabilities, there may also be 16-bit PC Card-only sockets in the system. From the client's point of view, whether the 16-bit PC Card is in a 16-bit PC Card-only socket or in a combined 16-bit PC Card/ CardBus PC Card socket, the Card Services programming interface is the same.

The CardBus PC Card Socket Services knows that the card which has just been inserted is a 16-bit PC Card by reading the adapter. After that point, Card Services continues its normal processing just as if the card were in a 16-bit PC Card-only socket. The only difference is the actual hardware interface, which is isolated to Socket Services. This processing consists of:

1. Card Services notifies its registered clients that a card has arrived.
2. The client walks the CIS via calls to Card Services.
3. If it recognizes the card, the client has the card configured by requesting resources such as memory windows from Card Services.
4. After the card is configured, the client may manipulate the card.

© 1999 PCMCIA/JEIDA

4.4 Fax/Modem CIS Design

4.4.1 Summary

This guideline describes the design of CIS for a single function 9600/2400 BPS Fax/Modem I/O 16-bit PC Card (not CardBus). The functional information required for a well defined CIS for a similar function on a multiple function PC Card or CardBus PC Card would be the same. However, the requirements for control tuples and device information tuples, etc. vary among these card types. Only the single function 16-bit PC Card is given in this example. How to use the tuple structure and what are appropriate values for field values are included in the examples.

4.4.2 Background

The purpose of the Card Information Structure is to provide a mechanism for software to be able to identify the function, origin, and capabilities of a PC Card. The idea is that when a card is inserted (or the machine is turned on with the card installed), an intelligent piece of code, which might be in the BIOS or run as a TSR or a device driver, would parse the CIS and perform the needed operations for card and system configuration. Being able to configure the card either at boot or after insertion is desirable since the host computer does not need to tie up resources before they are needed. It also allows the system to "gracefully reject" any card that it is not capable of using.

The discussion that follows is focused on a 9600/2400 Fax/Modem, but can be used for any type of I/O card. A Fax/Modem was selected because the extension tuples for this type of card were the first to be added to the ***PC Card Standard***. Using these function extension tuples along with the Rev 2.0 configuration tuples, a full descriptive guideline can be given. The 9600/2400 speed was selected for this guideline because it has become the generic minimum for Fax/Modems.

Some of the 9600/2400 Fax/Modem specific features that are documented in this guideline are:

9600 Fax speed

2400 Modem speed

16450 UART

All character, stop, and start bit possible configurations

Support for the US

V.22bis, Bell 212A, V.22A&B, V.21, Bell 103

V.29, MNP1-5, V.42, V.42bis, V.27ter

©1999PCMCIA/JEIDA

4.4.3 Guideline

The tuples covered in this example are enumerated in Table 4-1: Tuple Codes for Fax/Modem Card Example, below. A discussion of each tuple and some of its values follows.

Table 4-1: Tuple Codes for Fax/Modem Card Example

Data Code	Tuple Name	Description
01H	CISTPL_DEVICE	Device Information
15H	CISTPL_VERS_1	Level 1 Ver / Product Info
20H	CISTPL_MANFID	Manufacturer ID
21H	CISTPL_FUNCID	Function ID
22H	CISTPL_FUNCE	Function ID Extensions
1AH	CISTPL_CONF	Configurable Card
1BH	CISTPL_CE	Configuration Entry
14H	CISTPL_NO_LINK	No Link
FFH	CISTPL_END	End of Chain

4.4.3.1 The Device Information Tuple (01H)

The ***PC Card Standard*** implies that this tuple must be the first tuple located in Attribute memory address space. Besides being the first tuple, the Device ID tuple is also the most confusing of the required tuples for I/O cards. This is because it originated from the PCMCIA 1.0/JEIDA 4.0 publications which assumed that there was some type of device in common memory.

Table 4-2: Device Information Tuple

Data Code	Tuple Name	Description
01H	CISTPL_DEVICE	Device Information Tuple
03H	TPL_LINK	Link to Next Tuple
00H	DEVICE INFO1 (00 = DTYPE_NULL)	No Device in Common Memory
00H		Empty Block of 512 bytes
FFH		End of Device Info Field

As shown in Table 4-2: Device Information Tuple, the device information tuple contains a minimum of 5 bytes. The first two bytes contain the tuple code and the link to the next tuple in the list. The link is calculated starting with the next byte of the tuple after the link and ending at the last byte of the tuple. So for this example, the value placed in the TPL_LINK field is 03H.

The next two bytes contain the common memory device information for the first device. In the fax/modem card there is just one device, the Fax/Modem, but in future products, multi-function cards will exist and consequently additional DEVICE INFO fields will be required. Each of the functions requires 2 bytes of device information. The two bytes contain the device type code, whether a Write Protect Switch is included, what speed the device is, and how much address space it needs. A value of 00H is placed in the first device ID byte because there is no device in common memory (in this Fax/Modem example), so a null device type is selected.

The device size which is the second byte of the device information field is a 00H because that is the smallest amount allowed. With the value of 00H a window of 1 block of 512 bytes is defined.

Unfortunately, the byte does not allow for no memory, so you should specify the smallest one possible. The last byte is an FFH which marks the end of the device information tuple.

4.4.3.2 Level 1 Version/Product Information Tuple (15H)

The level 1 version/product information tuple is used to indicate the level of the card and also the card's manufacturer. The ***PC Card Standard*** describes two version levels, however, this tuple is used for Level 1 tuples only.

Table 4-3: Version Product Information Tuple

Data Code	Tuple Name	Description
15H	CISTPL_VERS_1	Level 1 Version / Product Information Tuple
1EH	TPL_LINK	Link to Next Tuple
04H	TPLLV1_MAJOR	Major Revision Number
01H	TPLLV1_MINOR	Minor Revision Number
50H	TPLLV1_INFO	P
43H		C
4DH		M
43H		C
49H		I
41H		A
00H		Terminator
46H		F
61H		a
78H		x
20H		space
43H		C
61H		a
78H		r
64H		d
00H		Terminator
46H		F
58H		X
33H		3
32H		2
31H		1
30H		0
00H		Terminator
41H		A
2DH		-
30H		0
00H		Terminator
FFH		End of List

©1999PCMCIA/JEIDA

The TPLLV1_MAJOR and _MINOR fields are defined in the PCMCIA 1.0/JEIDA 4.0 publications as having a value of 05H for _MAJOR and 00H for _MINOR in the PCMCIA 2.0/JEIDA 4.1 publications. This value should only be changed for your configuration if you are conforming to a later release of the ***PC Card Standard***.

The TPLLV1_INFO field is used to input the manufacturer, the product name and revision number. The specification is not specific about the exact format for this, however, the example above should be followed including the use of terminators and including the end of list (FFH) field at the end. Incidentally, the end of list field is not a requirement for all tuples, but it is required for this one. The values placed in this field are the ASCII hex values of the letters indicating the product name, revision number, vendor specific information, and manufacturer (PCMCIA).

4.4.3.3 Manufacturer's ID Tuple (20H)

This tuple was added to provide the ability for software to determine the origin and manufacturer of the PC Card as well as the card type.

Table 4-4: Manufacturers ID Tuple

Data Code	Tuple Name	Description
20H	CISTPL_MANFID	Manufacturing ID Tuple Device Information Tuple
04H	TPL_LINK	Link to Next Tuple
XXH	TPLMID_MANF	Manufacturer's ID Tuple (LSB)
XXH		Manufacturer's ID Tuple (MSB)
XXH	TPLMID_CARD	Company Specific Info
XXH		Company Specific Info

The TPLMID_MANF field is the unique ID code that is developed from the JEDEC Device ID code indicating the manufacturer. If a manufacturer has a JEDEC code, then the 16 bit PCMCIA ID has a 00H in the upper byte, and the code in the lower byte. If a manufacturer does not have a JEDEC code, PCMCIA can assign a company specific Manufacturer's ID Tuple.

The TPLMID_CARD field designates the card type. These two bytes are manufacturer specific identifiers. Each manufacturer will develop its own coding scheme for its card family.

4.4.3.4 Function ID Tuple (21H)

The purpose of this tuple is to provide information regarding the function of the card and system initialization information.

Table 4-5: Function ID Tuple

Data Code	Tuple Name	Description
21H	CISTPL_FUNCID	Function ID Tuple
02H	TPL_LINK	Link to Next Tuple
02H	TPLDFID_FUNCTION	Serial I/O Fax/Modem
01H	TPLFID_SYSINIT	Configure at Post

The TPLDFID_FUNCTION field specifies the function, or multiple functions the card can perform. The example Fax/Modem card performs as a serial device for both modem and fax functions and

thus has the value of 02H. A value of 0H indicates a multiple function card and this requires additional bytes to specify the individual functions. Currently the following values are:

0	Vendor Specific Multiple Function
1	Memory
2	Serial Port (both modem and fax cards)
3	Parallel Port (parallel printer port both uni- and bi-directional)
4	Fixed Disk
5	Video Adapter
6	LAN Adapter
7	AIMS
8	SCSI
9	Security
A	Instrument
B-FFH	Reserved

The TPLFID_SYSINIT field of the tuple indicates to the host when the card should be initialized. Only two values are defined today; a 01H indicates that the card should be initialized during POST, and a value of 02H indicates that the card contains expansion ROM which should be installed during system installation. Our example card does not contain ROM and consequently should have the value of 01H.

4.4.3.5 Modem Function Extension Tuple (22H)

The Modem Function Extension tuples document the capabilities of the fax or modem before the communications package tries to use it.

©1999PCMCIA/JEIDA

Table 4-6: Function Extension Tuples

Data Code	Tuple Name	Description
22H	CISTPL_FUNCE	Serial Port ID
04H	TPL_LINK	Link To Next Tuple
00H	TPLFE_TYPE	Serial Port
01H	TPLFE_UA	16450 UART
0FH	TPLFE_UC	All Parity Supported
7FH		All Stops and Characters Supported
22H	CISTPL_FUNCE	Modem Function Extension
09H	TPL_LINK	Link to Next Tuple
01H	TPLFE_TYPE	Modem as Discrete Device
1FH	TPLFE_FC	All Flow Control Methods
09H	TPLFE_CB	40 Character DCE Command Buffer
C8H	TPLFE_EB	200 Character DCE to DCE Buffer, LSB of LSW
00H		DCE to DCE Buffer MSB of LSW
00H		DCE to DCE LSB of MSW
C8H	TPLFE_TB	200 Character DTE to DCE Buffer, LSB of LSW
00H		DTE to DCE MSB of LSW
00H		DTE to DCE LSB of MSW
22H	CISTPL_FUNCE	Data Modem Extension
0CH	TPL_LINK	Link to Next Tuple
02H	TPLFE_TYPE	Modem
00H	TPLFE_UD	DTE to UART Max Data Rate MSB
80H		DTE to UART Max Data LSB (9600)
3BH	TPLFE_MS	Modulation V.22bis,Bell 212A, V.22A and B, V.21, Bell 103
00H		Not Supported
03H	TPLFE_EM	Error Correction using MNP 2-4 or V.42/LAPM
03H	TPLFE_DC	Data Compression using MNP5 or V.42bis
08H	TPLFE_CM	Command Protocol is MNP AT
07H	TPLFE_EX	Escape Mechanism is User Defined, +++, Break
00H	TPLFE_DY	No Data Encryption
00H	TPLFE_EF	No Caller ID
B5H	TPLFE_CD	ITU-T(CCITT) Country Code (USA)
22H	CISTPL_FUNCE	Fax Extension
08H	TPL_LINK	Link to Next Tuple
13H	TPLFE_TYPE	Fax Class 1
00H	TPLFE_UD	DTE to UART Max Data Rate MSB
80H		DTE to UART Max Data Rate LSB (9600)
06H	TPLFE_FM	ITU-T(CCITT) Modulation Supported V.29, V.27ter
00H	TPLFE_FY	Reserved for Future Encryption Definition
22H	TPLFE_FS	Fax features Polling and T.4
00H		Reserved
B5H	TPLFE_CF	ITU-T(CCITT) Country Code USA

© 1999 PCMCIA/JEIDA

Using these tuples will allow the host and the application software, which is CIS aware, to do away with the current set of questions that must be asked when configuring the operating environment.

The fax/modem tuple list contains three different function extension tuples. These are used to describe the three distinct areas of functionality of fax/modem products. The current extension tuple has been designed to accommodate one more function (not included in this example card) — Voice Encoding. The tuples start with the simplest function, a UART or Universal Asynchronous Receiver Transmitter. The serial port ID tuple is a 6 byte tuple. The functions that are included are the determination of the UART type, the start and stop characters supported, and the parity types.

There are three types of UARTs used for serial communication, they are the 8250, 16450 and the 16550. The fourth and fifth fields deal with the actual UART capabilities. The standard 16450 UART is capable of handling all data sizes (5, 6, 7, 8), stop characters (1, 1.5, 2) and parity types (even odd, mark, or space). The 16450 is the UART used in this example.

The Modem Interface tuple, which starts with the second CISTPL_FUNCE code, details the functionality of the modem . This tuple string describes the modem characteristics. This is a eleven byte tuple, starting with the standard code and link tuples. The next byte deals with the flow control methods of the modem. Flow control is the handshaking between two modems and is used to decide if the connection can support the sending/receiving of new information. A byte value of 1Fh corresponds to the support of all flow control methods including hardware and software flow control. This tuple also describes the chip set feature of a 40 character DCE Command Buffer, and 2 character buffers. The 40 character command buffer gives the tuple value of 09h because the formula used for calculation is to take the buffer size, divide by 4, then subtract a 1 (size = n, n/4-1). On our example card, the two character buffers are 200 character DCE to DCE and DTE to DCE buffers.

The Data Modem tuple is the third of the five CISTPL_FUNCE code tuples. This tuple describes the data rates, modulation schemes, error correction and detection, command protocols, and the ITU-T(CCITT) Country code. This tuple string is 14 bytes long, starting with the standard tuple code and link. The values for this string of data comes straight off of the modem chip set (internal to the card) data sheets, consequently, when developing the data for this tuple refer to the appropriate data sheets.

The Fax tuple is the final function extension tuple in this section. This tuple is a minimum of ten bytes, but must be replicated for each supported Service class. As with the Data modem tuple, the features described are the maximum data rate, modulation schemes, and country code.

4.4.3.6 Configuration Tuple (1AH)

The Configuration Tuple, Tuple 1AH, describes the general characteristics of PC Cards, for example, where the first configuration register is located and how many there are in total.

Table 4-7: Configurable Card Tuple

Data Code	Tuple Name	Description
1AH	CISTPL_CONF	Configuration Tuple
05H	TPL_LINK	Link to Next Tuple
01H	TPCC_SZ	Two bytes of Address
23H	TPCC_LAST	23H (Com4) is Last entry in Configuration Table
00H	TPCC_RADR	Configuration Register Base Address LSB
02H	TPCC_RADR	Configuration Register Base Address MSB
03H	TPCC_RMSK	Two Configuration Registers

©1999PCMCIA/JEIDA

The TPCC_SZ field indicates the number of bytes in the configuration registers base address, and the configuration register presence mask. The values of these registers are detailed in the bytes following the size field.

The TPCC_LAST field gives the value that corresponds to the final entry in the card configuration table. For communications cards that are being designed for use with PCs, this value will correspond to COM4. In this example a 23H corresponds to COM4.

The TPCC_RADR field indicates the address of the configuration registers in Attribute memory address space. At least one byte of address is required. If more than one byte is needed, the LSB should be presented first.

The TPCC_RMSK field gives the number of registers that start at the base address in Attribute memory address space. Each bit in the field corresponds to the presence or absence of a register. The host would use this information along with the value in the RADR field to determine the location of the card's configuration register set.

4.4.3.7 Configuration-Entry Tuple (1BH)

The Configuration Entry tuple, Tuple 1BH, defines the default configuration and its description. If the fax/modem is expected to be used in the PC Market, the Configuration tuple would need to be repeated for the other three Comm ports of the PC.

© 1999 PCMCIA/JEIDA

Table 4-8: Configuration Entry Tuple

Data Code	Tuple Name	Description
1BH	CISTPL_CE	Configuration Entry Tuple
15H	TPL_LINK	Link to Next Tuple
E0H	TCE_INDX	Default: COM1 (20H), Interface Follows
01H	TPCE_IF	I/O, Wait Supported
9DH	TPCE_FS	Feature Selection Byte
71H	TPCE_PD	Power Description
55H		V nominal = 5 v
86H		I Avg. = 138 mA
26H		Extension
86H		I Peak = 197 mA
61H		Extension
64H		I Power Down = 6 mA
FCH	TPCE_TD	Wait Scale Only
0CH		10 uS Max Wait Time
AAH	TPCE_IO	8 Bit I/O 10 Bit I/O Addr
60H		1 Range, I/O Addr = 2 bytes
F8H		Start of I/O Addr (LSB)
03H		Start of Addr (MSB)
07H		Length of Range =8
30H	TPCE_IR	Level Mode, Mask for INTs
BCH		IRQ7, IRQ5, IRQ4, IRQ3, IRQ2
86H		IRQ15, IRQ10, IRQ9
28H	TPCE_MI	Supports Power Down and Audio

As with all tuples, the configuration tuple begins with the tuple code and the link to the next tuple. After the link is the Table Index Byte. A value of E0H defines that this entry is the default entry, that a set of interface descriptions follow, and that a value of 20H must be written into the Configuration register to enable the card.

The Interface Description field, feature selection byte field, and Power Selection byte are straight forward and describe that the card is an I/O card, that it supports the Wait line of the 68 pin interface, and details that the following bytes will describe the Voltage, Current, Wait scale, address port, Power Down features and Audio features of the card.

The most confusing bytes of information in this section are the Power Description Structure Parameter Definitions. These are not required, but help the system to get a full description of the card. Since this is confusing, we will step through the description in detail. Table 4-9: Power Extension Fields for Tuple 1Bh through Table 4-11: Mantissa Values are the tables given in the specification that detail the power description structure.

Table 4-9: Power Extension Fields for Tuple 1BH

Byte	7	6	5	4	3	2	1	0
0	Ext	Mantissa				Exponent		
1	Ext	Extension						

©1999PCMCIA/JEIDA

Table 4-10: Current /Voltage Scales

Exponent	The Exponent of the Current and Voltage Values are given below:		
	Exponent	Current Scale	Voltage Scale
	0	100 nA	10 uV
	1	1 uA	100 uV
	2	10 uA	1 mV
	3	100 uA	10 mV
	4	1 mA	100 mV
	5	10 mA	1 V
	6	100 mA	10 V
	7	1 A	100 V

Table 4-11: Mantissa Values

Mantissa	The Mantissa of the value are given below:	
	0H	1
	1H	1.2
	2H	1.3
	3H	1.5
	4H	2
	5H	2.5
	6H	3
	7H	3.5
	8H	4
	9H	4.5
	AH	5
	BH	5.5
	CH	6
	DH	7
	EH	8
	FH	9

The nominal voltage is an example of a value not needing an extension. Looking at byte 0, the extension bit will be 0, since no extension is needed because the value can be defined with only one byte. The mantissa which occupies bits 6-3 will be an AH. This is seen from Table 4-11: Mantissa Values, where an AH represents the numerical value of 5. The Exponent which occupies bits 2-0 will be a 5 since the Voltage scale is 1V. Grouping all of the bits together gives a value of 55H.

The Peak Current of 197 mA needs an extension byte, byte 1, because the value to be described has 3 significant digits. As with the voltage, the designer must first look at byte 0. In this case bit 7 will be a 1, thus signifying that an extension byte follows. The mantissa will be a 0 because the value to be described is a 1 (see Table 4-11: Mantissa Values). Since we are looking at the 100 mA range, the Current scale for the exponent is 6. When put together the byte 0 value is 86H . This takes care of the 100's portion of the value. To get the remaining 97 mA defined, the extension byte is used. The value shown of 61H is the hex value for 97.

© 1999 PCMCIA/JEIDA

4.4.3.8 No Link Tuple (14H)

The NO_LINK tuple is an optional tuple that is used to help speed up tuple processing by indicating to the software that there will be no long link tuples in this chain. Long links are used when tuples are located past the 256 Byte limit of the link field. In particular, they are often used to indicate a tuple chain stored in common, rather than attribute memory.

Table 4-12: No Link Tuple

Data Code	Tuple Name	Description
14H	CISTPL_NO_LINK	No Link Tuple
00H	TPL_LINK	Link to Next Tuple

4.4.3.9 End of Tuple Chain (FFH)

The End of Tuple Chain is used to indicated the end of the tuple chain and has a value of FFH. This tuple appears at the end of the CIS.

Table 4-13: End of Tuple Chain

Data Code	Tuple Name	Description
FFH	CISTPL_END	End of Tuple Chain

©1999PCMCIA/JEIDA

4.5 Wireless CIS

4.5.1 Summary

The following is the suggested additions to the example Tuple requirements already proposed in other Guidelines.

4.5.2 Background

This is part of a series of proposed guidelines which describe sample CISs for a variety of cards.

4.5.3 Guideline

4.5.3.1 Wireless Modems

The tuples as defined in the Fax/Modem CIS Design are in general all that is needed for a wireless modem card. The differences of concern to a host between a conventional modem and wireless modem could be the maximum or peak power required when the RF unit is transmitting. Since provision for this information is already in the current modem tuple Guideline, no further definition is required here. Function extension tuples which define the particular RF technology supported by the card, i.e., ARDIS, CDPD, Mobitex, etc. could be of interest to applications (clients). If there is mutual interest in this, further modem extension tuples could be proposed.

4.5.3.2 Wireless LANS

As with the modems, the wireless LAN PC Card would require no additional tuples beyond that recommended for the non-wireless LAN. The wireless LAN PC Card should include a LAN function extension tuple for LAN_TECH_CODE with value of 7 specified and the LAN function extension tuple for LAN_MEDIA_CODE with the appropriate value representing the technology implemented.

4.5.3.3 Wireless Pagers

Since the wireless pager looks to the host like a serial port, no further tuples beyond that of the basic serial port are required. Once again, if it would be helpful to applications (clients), function extension tuples could be proposed that would further define the capabilities of the wireless pagers.

© 1999 PCMCIA/JEIDA

4.6 Sample PC Card ATA Tuple Options

4.6.1 Summary

Following is a summary of 16-bit PC Card ATA CIS considerations with actual samples for PC Card ATA tuple options. Although much of this information is also pertinent to CardBus PC Cards, the particular example does not deal with how CardBus PC Card ATA card's CIS should be designed.

4.6.2 Background

This is part of a series of proposed guidelines which describe sample CISs for a variety of cards.

4.6.3 Guideline

Following is the recommended minimum set of tuples for PC Card ATA cards. The tuples are listed in the order in which they would generally be encountered while scanning the CIS on the card. The tuples are defined in the ***Metaformat Specification*** and the ***PC Card ATA Specification***. In some cases, the tuple is only required if a card has certain characteristics (e.g., memory mapped ATA registers). In this case, this is noted in the table. If this guideline is found to differ with an explicit statement in the ***Metaformat*** or ***PC Card ATA Specifications***, those specifications shall be the controlling documents. Examples for the required and some optional configurations are included for the Configuration Table Entry tuple.

Note:
Shading is used to indicate portions of tuples which are dependent on the characteristics of a particular implementation of a PC Card ATA card.

©1999PCMCIA/JEIDA

4.6.3.1 CIS Usage for PC Card ATA Cards

Table 4-14: PC Card ATA Tuple Usage Chart

Tuple Code	Tuple Name	Tuple Defined in	Mode Tuple Used For			Notes
			Basic 3 PC Card ATA I/O Modes Only	Memory mapped PC Card ATA support	Non-PC Card ATA Functions	
01H	Device Info Tuple(Common)	PC Card Standard Metaformat	**Required** Null Device required if no memory space on card	**Required** Function Specific Regs	**Required** As needed to define card memory space	Specifies speed and size of Common Memory Areas (e.g., PC Card ATA Memory Mapped Registers) at 5 Volts **Vcc** when **WAIT#** signal is ignored by host. If no common memory space on card is accessible at 5 Volts **Vcc** it must be DTYPE_NULL and DSPEED_NULL.
1CH	Other Conditions Device Info (Common)	PC Card Standard Metaformat	**Required** Only if **WAIT#** or 3V **Vcc** used on card. Otherwise, not present.	**Required** Only if **WAIT#** or 3V **Vcc** used on card. Otherwise, not present.	As needed for other functions	Characteristics of Memory Address Spaces under other operating conditions as follows for Other Conditions Info Values: 0: Not Allowed 1: 5 Volt **Vcc**, Host uses **WAIT#** 2: 3 Volt **Vcc**, Host ignores **WAIT#** 3: 3 Volt **Vcc**, Host uses **WAIT#** Others: Reserved
18H	JEDEC ID (Common)	PC Card Standard Metaformat Values: PC Card ATA	None	**Required** DF-01 DF-02 DF-04 DF-08	As needed for other functions	Defines the type of device for each range of memory space defined in the Device ID Tuple. PC Card ATA JEDEC IDs: DF-01: No **VPP** Required DF-02: **VPP** on Write Required DF-04: **VPP** for Media Access DF-08: **VPP** Always Required
20H	Manufacturer's ID	PC Card Standard Metaformat	Recom-mended	Recom-mended	Recom-mended	Recommended for all cards to identify the Specific Manufacturer and Product in the card. Must not be used for general card recognition. Host should use Function ID and Function Extension tuples instead.
15H	Level 1 Version / Product	PC Card Standard Metaformat	Recom-mended	Recom-mended	Recom-mended	Recommended for all cards to display product identification strings in ASCII to end users. Must not be used by host software to determine card function unless standard methods have failed. Host should use Function ID and Function Extension Tuples.

Tuple Code	Tuple Name	Tuple Defined in	Mode Tuple Used For			Notes
			Basic 3 PC Card ATA I/O Modes Only	Memory mapped PC Card ATA support	Non-PC Card ATA Functions	
21H	Function ID	PC Card Standard Metaformat	**Required** Disk Function Value = 4.	**Required** Disk Function Value = 4.	As needed by function	Disk Function ID is required. Function Extension following Disk Function ID further refine product characteristics. Host Software must use this tuple with Function Extension tuples to determine that card has PC Card ATA function. Typically POST bit is set in System Initialization byte to request system to initialize card for possible boot.
22H	Function Extension following Disk Function ID tuple: TPLFE_TYPE = 01H TPLFE_DATA = 01H	PC Card Standard Metaformat, PC Card ATA Spec	Required	Required	Not used.	Tuple defines the PC Card Disk Interface Type is supported on the card.
22H	Function Extension following Disk Function ID tuple: TPLFE_TYPE = 02H TPLFE_DATA = as needed by card	PC Card ATA Spec	Strongly Recom-mended for pre-PC Card Standard, Feb. 1995. Required for PC Card Standard, Feb. 1995 and Dual Drive Card.	Strongly Recom-mended for pre-PC Card Standard, Feb. 1995. Required for PC Card Standard, Feb. 1995 and Dual Drive Card.	Not Used.	PC Card ATA Features Description for first (or only) drive on card.
22H	Function Extension following Disk Function ID tuple: TPLFE_TYPE = 03H TPLFE_DATA = as needed by card	PC Card ATA Spec	Required for Dual Drive Card. Otherwise, Not Present.	Required for Dual Drive Card. Otherwise, Not Present.	Not Used.	PC Card ATA Features Description for second drive on card.
1AH	Card Configuration	PC Card Standard Metaformat	Required	Required	As needed by other functions.	Specifies location of Card Configuration Option Register and presence of other Card Configuration Registers. Must be used by host to determine location of configuration registers.

©1999PCMCIA/JEIDA

Tuple Code	Tuple Name	Tuple Defined in	Mode Tuple Used For			Notes
			Basic 3 PC Card ATA I/O Modes Only	Memory mapped PC Card ATA support	Non-PC Card ATA Functions	
1BH	Configuration Entry with Configuration Index=UU (00H as used in example is strongly recommended as an index value for cards supporting Memory Mapped PC Card ATA mode.) Specific Index values are at the discretion of the card vendor.	PC Card Standard	**Not Used**	**Required**	Additional information as needed by concurrent additional card function.	Defines card operation when card emerges from reset with the memory mapped only interface active. The Memory Mapped PC Card ATA mode should be active in this state if the mode is supported by the card. Card support in this configuration for MWAIT on Common Memory, Ready, and Write Protect Switch functions in memory mode can be indicated. Power and Memory Space requirements are defined here. I/O space and Interrupts may not be defined because no I/O exists in the Memory Only Interface.
1BH	Configuration Entry with Configuration Index = VVH (01H in example)	PC Card Standard Metaformat	**Required**	**Not Used** for Memory mode, but **Required** for I/O modes also supported on card.	Additional function	Defines card operation when card is placed in PC Card ATA Contiguous I/O mode. Card support in this configuration for MWAIT on Common Memory, Ready, and Write Protect Switch functions in I/O mode can be indicated. Power, I/O Space (4 address lines decode only), and Interrupt (any) requirements are defined here.
1BH	Configuration Entry with Configuration Index = XXH. (02H in example)	PC Card Standard Metaformat	**Required**	**Not Used** for Memory Mode, but **Required** for I/O modes also supported on card.	May contain additional information as needed by additional card function.	Defines card operation when card is placed in PC Card ATA Primary I/O mode. Card support in this configuration for MWAIT on Common Memory, Ready, and Write Protect Switch functions in I/O mode can be indicated. Power, I/O Space (10 address lines decoded, Windows 1F0-1F7 and 3F6-3F7) requirements and Interrupt (usually IRQ 14 for AT BIOS compatibility) recommendations are defined here.

© 1999 PCMCIA/JEIDA

Tuple Code	Tuple Name	Tuple Defined in	Mode Tuple Used For			Notes
			Basic 3 PC Card ATA I/O Modes Only	Memory mapped PC Card ATA support	Non-PC Card ATA Functions	
1BH	Configuration Entry with Configuration Index = YYH. (03H in example)	PC Card Standard Metaformat	Required	Not Used for Memory Mode, but Required for I/O modes also supported on card.	May contain additional information as needed by additional card function.	Defines card operation when card is placed in PC Card ATA Secondary I/O mode. Card support in this configuration for MWAIT on Common Memory, Ready, and Write Protect Switch functions in I/O mode can be indicated. Power, I/O Space (10 address lines decoded, Windows 170-177 and 376-377) requirements and Interrupt (usually IRQ 14 for AT BIOS compatibility) recommendations are defined here.
1BH	Configuration Entry with Configuration Index = ZZH. (not shown in example)	PC Card Standard Metaformat	Required for each additional config-uration present.	Required for each additional config-uration present.	May contain additional information as needed by additional card function.	Set of configurations with unique Configuration Index values are used to alternative PC Card ATA configurations. These configurations may differ from the previous four configurations in requirements for Power, I/O and Memory Space decoding,
14H	No-Link Control	PC Card Standard Metaformat	Strongly Recom-mended.	Strongly Recom-mended.	Strongly Recom-mended.	Prevents tuple processing from doing an implied long link jump to common memory. Should not appear if an explicit Long Link tuple is present.
FFH	End of List	PC Card Standard Metaformat	Required unless FFh link used.	Required unless FFh link used	Required unless FFh link used	Terminates each tuple list on the card.

©1999PCMCIA/JEIDA

4.6.3.2 Sample Tuples for PC Card ATA Cards

Table 4-15: Sample Device Info Tuple–01H Required on PC Card ATA Cards

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	01H	CISTPL_DEVICE								Device Info Tuple	Tuple Code
2: 1 Byte	??H	TPL_LINK									Link to next tuple
3: 0 or more Bytes	??H	Device ID Structures for Other Devices or Null Devices for address offset computation.								For Memory Mapped PC Card ATA. When PC Card ATA registers are not at address offset 0, Device ID Structures preceding this byte must have sizes which sum to the offset address of the PC Card ATA registers.	Device ID Structures
4: 1 Byte	??H	Device Type				**WP**	Dev Speed				Device Type, WPS, Speed when **WAIT#** not supported
	DFH	DH = Func Spec				1	7 = Extend			Func Specific, No WPS, ext speed	PC Card ATA Regs
	D9H	DH = Func Spec				1	1 = 250 nsec			Func Specific, No WPS, speed 250	PC Card ATA Regs
	00H	0H = Null				0	0 = Null			No Memory Mapped PC Card ATA	No Device
5: 0 or 1 Byte	72H 42H	0 0 X	EH = 7.0 8H = 3.5 Spd Mantis				Spd Expo 2H = 100 nsec			700 nsec no wait 350 nsec if no wait	Extended Speed (present only if speed is extended)
6: 1 Byte	??H	Addr Units - 1					Unit Size				Device Size Byte for
	01H	1 unit of					2 K bytes			2 Kilobyte Address Space	PC Card ATA Regs
7: 0 or more Bytes		Device ID Structures for Common Memory Space used by Other Card Functions								When other memory devices exist in common memory, additional Device ID Structures may occur before the List End Marker.	Device ID Structures
8: 1 Byte	FFH	List End Marker								End of Devices	End Marker

© 1999 PCMCIA/JEIDA

Table 4-16: Sample Other Conditions Device Info Tuple–1CH Required only if 3V VCC or WAIT# signal is supported by card for Common Memory cycles

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1CH	CISTPL_DEVICE_OC								Other Conditions Device Info Tuple	Tuple Code
2: 1 Byte	??H	TPL_LINK									Link to next tuple
3: 1 Byte	0?H	E	R	R	R	R	R	3	M	Other Conditions Which Apply: MWAIT = 1: **WAIT#** Signal Used 3VCC = 1: 3 Volt **Vcc** Used EXT = 1: Extension Bytes Follow RSVD: Reserved, must be 0	Other Conditions Info Field
	01H 02H 03H	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 1 1	1 0 1	**WAIT#** Signal Used, 5 V **Vcc** **WAIT#** Signal Not Used, 3 V **Vcc** **WAIT#** Signal Used, 3 V **Vcc** all others reserved	
4: 0 or more Bytes	??H	Device ID Structures for Other Devices or Null Devices for address offset computation								For Memory Mapped PC Card ATA. When PC Card ATA registers are not at address offset 0, Device ID Structures preceding this byte must have sizes which sum to the offset address of the PC Card ATA registers.	Device ID Structures which apply for **WAIT#** used in common memory. One to one correspondence with Device ID Tuple fields.
5: 1 Byte	??H	Device Type				**WP**	Dev Speed				Device Type, WPS, Speed when **WAIT#** is supported.
	D9H	DH = Func Spec				1	1 = 250 ns			Func Specific, No WPS, speed 250	PC Card ATA Regs
	00H	0H = Null				0	0 = Null			No Memory Mapped PC Card ATA	No Device
6: 1 Byte	??H	Addr Units - 1					Unit Size				Device Size Byte for
	01H	1 unit of					2 K bytes			2 Kilobyte Address Space	PC Card ATA Regs
7: 0 or more Bytes		Device ID Structures for Common Memory Space used by Other Card Functions								When other memory devices exist in common memory, additional Device ID Structures may occur before the List End Marker.	Device ID Structures. One to one correspondence with Device ID Tuple fields.
8: 1 Byte	FFH	List End Marker								End of Devices	End Marker

©1999PCMCIA/JEIDA

Table 4-17: Sample JEDEC ID Tuple–18H
Required for PC Card ATA Cards supporting Memory Mapped PC Card ATA Registers

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
0: 1 Byte	18H	CISTPL_JEDEC_C								JEDEC ID Common Mem	Tuple Code
1: 1 Byte	??H 02H	TPL_LINK 02H = Length for a single JEDEC ID								Link is 2 bytes for PC ATA Only in memory at offset 0.	Link Length
2: 0 or more Bytes (Even)	??H	JEDEC ID Codes for Device ID Structures preceding the PC Card ATA Registers description in the Device Info Tuple.								Each Entry is 2 bytes long.	
3: 1 Byte	DFH	PCMCIA's Manufacturer's JEDEC ID code. DFH = PCMCIA								Manufacturer's ID Code	Byte 1, JEDEC ID of PC Card ATA Registers in common memory
4: 1 Byte	??H 01H 02H 04H 08H	PCMCIA JEDEC Device Code 01H = No **VPP** Used PC Card ATA 02H = **VPP** on Write PC Card ATA 04H = **VPP** on Media Access 08H = **VPP** Continuous PC Card ATA								Second Byte of JEDEC ID. Typically, as in this case, the Device Specific Code for a 1 Byte Manufacturer Code. Each of these codes specifies PC Card ATA with different **VPP** protocol.	Byte 2, JEDEC ID of PC Card ATA registers in common memory
5: 0 or more Bytes (Even)	??H	JEDEC ID Codes for Device ID Structures after the PC Card ATA Registers description in the Device Info Tuple.								Each Entry is 2 bytes long.	

Table 4-18: Sample Manufacturer's ID Tuple–20H
Optional but recommended, for PC Card ATA Devices

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	20H	CISTPL_MANFID								Manufacturer's ID Tuple	Tuple Code
2: 1 Byte	04H									Link is 4 bytes	Link Length
3: 1 Byte	??H	Request 16 bit Code from PCMCIA or use your JEDEC assigned Manufacturer's ID code if you have one								Low Byte of Manufacturer's ID Code assigned by PCMCIA	Low Byte of Manufacturer's ID Code
3: 1 Byte	??H or 00H	Code of 00 used when manufacturer uses a 1 Byte JEDEC assigned Manufacturer's code								High Byte of Manufacturer's ID Code assigned by PCMCIA	High Byte of Manufacturer's ID Code
4: 1 Byte	??H	Low Byte of 16 bit Product Code assigned by Manufacturer								Manufacturer's code for PC Card ATA card of appropriate version	Low Byte of Product Code
5: 1 Byte	??H	High Byte of 16 bit Product Code assigned by Manufacturer								Manufacturer's code for PC Card ATA card of appropriate version.	High Byte of Product Code

© 1999 PCMCIA/JEIDA

Table 4-19: Sample Level 1 Version / Product Info Tuple–15H
Recommended

Order / Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	15H	CISTPL_VER_1								Level 1 version / product info	Tuple Code
2: 1 Byte	??H									Link to next tuple is ?? bytes	Link Length
3: 1 Byte	04H	TPPLV1_MAJOR								PCMCIA 2.0 /JEIDA 4.1	Major Version
4: 1 Byte	01H	TPPLV1_MINOR								PCMCIA 2.0 / JEIDA 4.1	Minor Version
5: 1 or more Bytes	56H	Manufacturer Name String (ASCIIZ)								'V'	Info String 1
	65H									'e'	
	6EH									'n'	
	64H									'd'	
	6FH									'o'	
	72H									'r'	
6: 1 Byte	00H	End of Manufacturer Name String								Null terminator	
7: 1 or more Bytes	50H	Product Name String (ASCIIZ)								'P'	Info String 2
	43H									'C'	
	20H									' '	
	43H									'C'	
	41H									'A'	
	52H									'R'	
	44H									'D'	
	20H									' '	
	41H									'A'	
	54H									'T'	
	41H									'A'	
8: 1 Byte	00H	End of Product Name String								Null terminator	
9: 0, 2 or more bytes		Optional additional Strings with Null Terminator for each one								Vendor Specific Strings	
10: 1 Byte	FFH	End of List Marker								FFH List terminator	No More Info Strings

Table 4-20: Function ID Tuple, Disk Function–21H
Required for PC Card ATA Cards

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
0: 1 Byte	21H	CISTPL_FUNCID								Function ID tuple	Tuple Code
1: 1 Byte	02H	TPL_LINK								This tuple has 2 info bytes	Link Length
2: 1 Byte	04H	TPLFID_FUNCTION = 04H								Disk Function Code, May be silicon, May be removable	PC Card Function Code
3: 1 Byte	0?H	Reserved = 0						R	P	ROM Present for BIOS on Card POST Configure card at Power On	System Initialization Byte
	00H	0						0	0	RP = 00: No ROM, No POST config.	
	01H	0						0	1	RP = 01: Card has no BIOS ROM but should be configured at POST	

©1999PCMCIA/JEIDA

Table 4-21: Disk Function Extension Tuple–Interface Type Required for PC Card ATA Cards

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
0: 1 Byte	22H	CISTPL_FUNCE								Function Extension tuple	Tuple Code
1: 1 Byte	02H									this tuple has 2 info bytes	Link Length
2: 1 Byte	01H	Disk Function Extension Tuple Type								Disk Interface Type	Extension Tuple Type for Disk
3: 1 Byte	01H	Disk Interface Type								PC Card ATA Interface	Interface Type

Table 4-22: Sample Disk Function Extension–PC Card ATA Parameters Tuple Strongly Recommended. Required for PC Card Standard Feb. 1995 and Dual Drive PC Card ATA

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
0: 1 Byte	22H	CISTPL_FUNCE								Function Extension tuple	Tuple Code
1: 1 Byte	03H									this tuple has 3 info bytes	Link Length
2: 1 Byte	02H or 03H	Disk Function Extension Tuple Type								Basic PC Card ATA Features Function Extension tuple	Extension Tuple Type for Disk Function
	02H	02H								Single Drive Features	
										or Master of Dual Drive Features	
	03H	03H								Slave of Dual Drive Features	
3: 1 Byte		R	R	R	D	U	S	V		V=0: No **VPP** Required V=1: **VPP** on Modify Media V=2: **VPP** on any operation V=3: **VPP** continuous S=1: Silicon; =0: Rotating U=1: ID Drive Mfg/SN Unique R: Reserved, Must Be 0.	Basic ATA Option Parameters Byte 1
										D=1: Dual Drives on Card (Pending)	
	04H 0CH 0DH	0 0 0	0 0 0	0 0 0	0 0 0	1 1 1	0 1 1	0 0 1		Rotating, No **VPP**, Unique Serial # Silicon, No **VPP**, Unique Serial # Silicon, **VPP** to Write, Unique Serial	Sample 1 Sample 2 Sample 3
4: 1 Byte	??H	R	I	E	N	P3	P2	P1	P0	P0 Sleep Mode Supported P1 Standby Mode Supported P2 Idle Mode Supported P3 Drive Auto Power Control N Some Config Excludes 3X7 E Index Bit is Emulated I Twin **IOIS16#** Data Reg Only R: Reserved, Must Be 0	Basic ATA Options Parameters Byte 2
	0FH	0	0	0	0	1	1	1	1	All power modes supported but power commands are not needed to minimize power. No Configs exclude I/O port 3F7H/377H, Index bit is not emulated, Twin Card not implemented, **IOIS16#** handling is not specified.	

© 1999 PCMCIA/JEIDA

Table 4-23: Sample Configuration Tuple
Required for all PC Card ATA cards.

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1AH	CISTPL_CONF								Configuration Tuple	Tuple Code
2: 1 Byte	??H									Link Length is N bytes	Link to next tuple
3: 1 Byte	??H	RFS		RMS				RAS		RFS Bytes in Reserved Field RMS Bytes in Reg Mask - 1 RAS Bytes in Base Addr - 1	Size of fields byte (TPCC_SZ)
	01H	0		0				1		No Reserved Field, 1 Byte Register Mask, 2 byte Config Base Address	
4: 1 Byte	??H	TPCC_LAST								Entry with Config Index of ?? is final entry in table	last entry of configuration table
5: 1-4 Bytes	00H	TPCC_RADR (lsb)								Configuration Registers are	Location of Config
	02H	TPCC-RADR (msb)								located at 200H in Reg Space	Registers
6: 1 Byte (when more configuration Registers are defined, this field may be longer	??H	R	R	R	E	S	P	C 1	I 1	I Configuration Index Required for All PC Card ATA Cards. C Configuration and Status Required only if power-down, audio, or Signal on Change is supported. P Pin Replacement Required only if Ready, Write Protect, or BVDs supported in I/O mode. S Socket and Copy Required only for cards supporting Twin Card option. R Reserved for future use.	TPCC_RMSK
	07H	0	0	0	0	0	1	1	1	First 3 Configuration Registers present (No Socket and Copy)	
	0FH	0	0	0	0	1	1	1	1	First 4 Configuration Registers present	

©1999PCMCIA/JEIDA

Table 4-24: Sample Configuration Entry Tuple for Memory Mapped I/O PC Card ATA Configuration
Required if Memory Mapped ATA Registers Supported

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1BH	CISTPL_CFTABLE_ENTRY								Configuration Table Entry Tuple	Tuple Code
2: 1 Byte	??H									Link to next tuple is N bytes. Also limits size of this tuple to N+2 bytes.	Link to next tuple
3: 1 Byte	??H	I	D	Configuration Index						I=1: Interface Byte Follows D=1: Independent "Default" Entry Card Configuration Index for Entry	TPCE_INDX
	C0H	1	1	0						Interface Byte Follows, Default Entry, Configuration Index = 0	**PC Card ATA Memory Mapped I/O Configuration**.
4: 1 Byte	C0H	W	R	P	B	Interface Type				**WAIT**=1: Wait Used on Mem Cycles **READY**=1: Ready Active **WP**=1: Write Prot Switch Active **BVD**=1: **BVD1** and **BVD2** Active IF Type=0: Memory Only I/F IF Type=1: I/O and Memory I/F IF Type=2-3,8-F: Reserved IF Type=4-7: Custom Interface	TPCE_IF
	C0H	1	1	0	0	0				Mem Interface (0); Bvd's and wProt not used; Ready active and Wait used for memory cycles.	TPCE_IF
5: 1 Byte		M	MS		IR	IO	T	P		**P** Power info type **T** Timing info present **IO** I/O port info present **IR** Interrupt info present **MS** Mem space info type **M** Misc info byte(s) present	TPCE_FS
	A1H	1	1H		0	0	0	1H		Has **Vcc**, Mem Space and Misc Info	**Vcc** Only Card
	A2H	1	1H		0	0	0	2H		Has **Vcc**, **VPP**, Mem Space and Misc Info present	**Vcc**, **VPP** Card
6: 1 Byte	01H	R	DI	PI	AI	SI	HV	LV	NV	Nominal Voltage Only Follows **NV** Nominal Voltage Info present **LV** Minimum Voltage Info present **HV** Maximum Voltage Info present **SI** Static Current Info present **AI** Average Current Info present **PI** Peak Current Info present **DI** Power Down Current Info present **R** Reserved, must be 0.	Power Parameters for **Vcc**
	01H	0	0	0	0	0	0	0	1	Nominal Voltage Only Follows	
	11H	0	0	0	1	0	0	0	1	Nom Voltage, Average Current	

© 1999 PCMCIA/JEIDA

Table 4-24: Sample Configuration Entry Tuple for Memory Mapped I/O PC Card ATA Configuration (Continued)
Required if Memory Mapped ATA Registers Supported

7: 1 or more bytes present only if NV=1	55H	X 0	Mantissa AH= 5.0				Exponent 5H= 1 V				**Vcc** Nominal is 5 Volts	**Vcc** Nominal Value
8: 1 or more bytes present only if AI=1	4EH 1EH	X 0 0	Mantissa 9H= 4.5 3H= 1.5				Exponent 6H= 100mA 6H= 100mA				Icc Average is 450 mA Icc Average is 150 mA	Icc Average Value
9: 1 Byte. Present only if TPCE_FS Power = 2.	??H	R	DI	PI	AI	SI	HV	LV	NV		See Above	Power Parameters for **VPP**
	01H	0	0	0	0	0	0	0	1		Nominal Voltage Only Follows	
10: 1 Byte. Present only if TPCE_FS Power = 2.	0EH	X 0	Mantissa 1H = 1.2				Exponent 6H = 10 V				**VPP** Nominal is 12 Volts	**VPP** Nominal Value
11: 2 Bytes	08H	Length in 256 bytes pages (lsb)									Length of Mem Space is 2 KB	TPCE_MS Length
	00H	Length in 256 byte pages (msb)									Starts at 0 on card	
12: 1 Byte	20H	X	R	P	RO	A	T				Twin Cards Allowed Audio Supported Read Only Mode Power Down Supported Reserved X More Misc Fields Bytes	TPCE_MI
09AH	20H	0 0	0 0	1 0	0 0	0 0	0 1				Power-Down, but no Twin Card. Twin Card: 2 Cards as Master/Slave	TPCE_MI

©1999PCMCIA/JEIDA

Table 4-25: Sample Contiguous I/O Mapped ATA Registers Configuration Entry Tuple Required for PC Card ATA Devices

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1BH	CISTPL_CFTABLE_ENTRY								Configuration Table Entry Tuple	Tuple Code
2: 1 Byte	??H									Link to next tuple is N bytes. Also limits size of this tuple to N+2 bytes.	Link to next tuple
3: 1 Byte	??H	I	D	Configuration Index						I = 1: Interface Byte Follows D = 1: Default Table Entry Conf Index: Index Value written to Card Config Option to select this entry	TPCE_INDX
	??H C1H	1	1	??H 01H						Default Entry, Interface specified in next byte Card Configuration 1	**I/O Mapped Contiguous 16 registers Configuration.**
4: 1 Byte		W	R	P	B	Interface Type				See Mem Mapped Sample Tuple	TPCE_IF
	41H	0	1	0	0	1				I/O Interface (); Bvd's and wProt not used; **READY** active and **WAIT#** not used for memory cycles.	
5: 1 Byte		M	MS		IR	IO	T	P		See Mem Mapped Sample Tuple	TPCE_FS
	99H 9AH	1 1	0H 0H		1 1	1 1	0 0	1H 2H		Has **Vcc**, I/O, IRQ and Misc Info Has **Vcc**, **VPP**, I/O, IRQ and Misc Info present	**Vcc** Only Card **Vcc**, **VPP** Card
6: 1 Byte	??H	R	D I	PI	AI	SI	H V	LV	N V	See Mem Mapped Sample Tuple	Power Parameters for **Vcc**
	01H 11H	0 0	0 0	0 0	0 1	0 0	0 0	0 0	1 1	Nominal Voltage Only Follows Nom Voltage, Average Current	
7: 1 or more bytes present only if NV=1	55H	X 0	Mantissa AH= 5.0				Exponent 5H= 1 V			**Vcc** Nominal is 5 Volts	**Vcc** Nominal Value
8: 1 or more bytes present only if AI=1	4EH 1EH	X 0 0	Mantissa 9H= 4.5 3H= 1.5				Exponent 6H= 100mA 6H= 100mA			Icc Average is 450 mA Icc Average is 150 mA	Icc Average Value
9: 1 Byte. Present only if TPCE_FS Power = 2	??H	R	D I	PI	AI	SI	H V	LV	N V	See Mem Mapped Sample Tuple	Power Parameters for **VPP**
	01H	0	0	0	0	0	0	0	1	Nominal Voltage Only Follows	
10: 1 Byte. Present only if TPCE_FS Power = 2	0EH	X 0	Mantissa 1H = 1.2				Exponent 6H = 10 V			**VPP** Nominal is 12 Volts	**VPP** Nominal Value

Table 4-25: Sample Contiguous I/O Mapped ATA Registers Configuration Entry Tuple Required for PC Card ATA Devices (Continued)

11: 1 Byte		R	S	E	IO AddrLines					IO AddrLines: #lines decoded Eight bit only hosts supported Sixteen bit hosts supported Range (I/O) follows	TPCE_IO
	64H	0	1	1	4					Supports both 8 and 16 bit I/O hosts. 4 Address lines and no range so 16 registers and host must do all selection decoding.	
12: 3 Bytes		S	P	L	M	Level or Mask				Share (IRQ) Logic Active	TPCE_IR
					0	IRQ Level				Pulse Mode IRQ Supported	TPCE_IR
					1	V	B	I	N	Level Mode IRQ Supported M: Bit Mask of IRQs Present IRQ Level (single) Vendor Unique IRQ Bus Error IRQ IO Check IRQ Non-Maskable IRQ	
	F0H	1	1	1	1	0	0	0	0	IRQ Sharing Logic Active in Card Control & Status Register, Pulse and Level Mode Interrupts supported, Recommended IRQ's any of 0 through 15 (F)	
	FFH	1	1	1	1	1	1	1	1	IRQ Levels to be routed 0 - 15 recommended. Bit 0 = IRQ0	TPCE_IR Mask Extension Byte 1
	FFH	1	1	1	1	1	1	1	1	Recommended routing to any "normal, maskable" IRQ. Bit 0 = IRQ8	TPCE_ IR Mask Extension Byte 2
13: 1 Byte	??H	X	R	P	RO	A	T			See Mem Mapped Sample Tuple	TPCE_MI
	20H	0 0	0 0	1 0	0 0	0 0	0 1			Power-Down, but no Twin Card. Twin Card: 2 Cards as Master/Slave without Power Down	TPCE_MI

©1999PCMCIA/JEIDA

Table 4-26: Sample ATA Primary I/O Mapped Configuration Entry Tuple Required for PC Card ATA Devices

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1BH	CISTPL_CFTABLE_ENTRY								Configuration Table Entry Tuple	Tuple Code
2: 1 Byte	??H									Link to next tuple is N bytes. Also limits size of this tuple to N+2 bytes.	Link to next tuple
3: 1 Byte		I	D	Configuration Index						I = 1: Interface Byte Follows D = 1: Default Table Entry Conf Index: Index Value written to Card Config Option to select this entry	TPCE_INDX
	??H 82H	? 1	0 0	??H 02H						Non Default Entry: Dependent on Previous Default Entry (Contiguous I/O). All unspecified fields taken from specified in next byte Sample Configuration Index = 2	**ATA Primary Address 1F0-1F7, 1F6-3F7 Configuration.**
4: 1 Byte		W	R	P	B	Interface Type				See Mem Mapped Sample Tuple	TPCE_IF
	41H	0	1	0	0	1				I/O Interface (); Bvd's and wProt not used; **READY** active and **WAIT#** not used for memory cycles.	
5: 1 Byte		M	MS		IR	IO	T	P		See Mem Mapped Sample Tuple	TPCE_FS
	18H	0	0H		1	1	0	0H		Has Power, Timing, Mem Space and Misc Info taken from contiguous configuration. I/O and IRQ Info specified here.	
6: 1 Byte	E4H	R	S	E	IO AddrLines					See Contiguous I/O Sample Tuple	TPCE_IO
	EAH	1	1	1	0aH					Supports both 8 and 16 bit I/O hosts. 10 Address lines and with range(s)	
7: 1 Bytes	??H	LS		AS		N Ranges				N Ranges: Number of Address Ranges following minus 1 AS: Size of Addresses 0: No Addresses Present 1: 1 Byte (8 bit) Addresses 2: 2 Byte (16 bit) Addresses 3: 4 Byte (32 bit) Addresses LS: Size of Lengths 0: No Lengths Present 1: 1 Byte (8 bit) Lengths 2: 2 Byte (16 bit) Lengths 3: 4 Byte (32 bit) Lengths	
	61H	1		2		1				2 Address Ranges with 2 byte addresses and 1 byte lengths	
8: 2 Bytes	F0H	1F0 (LSB) = F0H								First I/O Base Address (LSB)	First I/O Range Addr
	01H	1F0 (MSB) = 01H								First I/O Base Address (MSB)	First I/O Range Addr
9: 1 Byte	07H	8 Byte Length - 1 = 07H								First I/O Length minus 1	First I/O Range Len

© 1999 PCMCIA/JEIDA

Table 4-26: Sample ATA Primary I/O Mapped Configuration Entry Tuple Required for PC Card ATA Devices (Continued)

10: 2 Bytes	F6H	3F6 (LSB) = F6H					Second I/O Base Address (LSB)	2nd I/O Range Addr
	03H	3F6 (MSB) = 03H					Second I/O Base Address (MSB)	2nd I/O Range Addr
11: 1 Byte	01H	2 Byte Length - 1 = 01H					Second I/O Length minus 1	2nd I/O Range Len
12: 1 Bytes	??H	S	P	L	M	IRQ Level	See Contiguous I/O Sample Tuple	TPCE_IR
	EEH	1	1	1	0	EH	IRQ Sharing Logic Active in Card Control & Status Register, Pulse and Level Mode Interrupts supported, recommended IRQ's IRQ14 for ATA Compatibility	

©1999PCMCIA/JEIDA

Table 4-27: Sample ATA Secondary I/O Mapped Configuration Entry Tuple Required for PC Card ATA Devices

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	1BH	CISTPL_CFTABLE_ENTRY								Configuration Table Entry Tuple	Tuple Code
2: 1 Byte	??H									Link to next tuple is N bytes. Also limits size of this tuple to N+2 bytes.	Link to next tuple
3: 1 Byte	??H	I	D	Configuration Index						I = 1: Interface Byte Follows D = 1: Default Table Entry Conf Index: Index Value written to Card Config Option to select this entry	TPCE_INDX
	??H	?	0	??H						Non Default Entry: Dependent on Previous Default Entry (Contiguous I/O). All unspecified fields taken from specified in next byte	**ATA Secondary Address 170-177, 176-377 Configuration.**
	83H	1	0	03H						Sample Configuration Index = 2	
4: 1 Byte	41H	W	R	P	B	Interface Type				See Mem Mapped Sample Tuple	TPCE_IF
	41H	0	1	0	0	1				I/O Interface (); Bvd's and wProt not used; **READY** active and **WAIT#** not used for memory cycles.	
5: 1 Byte		M	MS		IR	IO	T	P		See Mem Mapped Sample Tuple	TPCE_FS
	18H	0	0H		1	1	0	0H		Has Power, Timing, Mem Space and Misc Info taken from contiguous configuration. I/O and IRQ Info specified here.	
6: 1 Byte	E4H	R	S	E	IO AddrLines					See Contiguous I/O Sample Tuple	TPCE_IO
	EAH	1	1	1	0AH					Supports both 8 and 16 bit I/O hosts. 10 Address lines and with range(s)	
7: 1 Bytes	??H	LS		AS	N Ranges					See ATA Primary I/O Tuple Sample	
	61H	1		2	1					2 Address Ranges with 2 byte addresses and 1 byte lengths	
8: 2 Bytes	70H	170 (LSB) = 70H								First I/O Base Address (LSB)	First I/O Range Addr
	01H	170 (MSB) = 01H								First I/O Base Address (MSB)	First I/O Range Addr
9: 1 Byte	07H	8 Byte Length - 1 = 07H								First I/O Length minus 1	First I/O Range Len
10: 2 Bytes	76H	376 (LSB) = 76H								Second I/O Base Address (LSB)	2nd I/O Range Addr
	03H	376 (MSB) = 03H								Second I/O Base Address (MSB)	2nd I/O Range Addr
11: 1 Byte	01H	2 Byte Length - 1 = 01H								Second I/O Length minus 1	2nd I/O Range Len
12: 1 Bytes	??H	S	P	L	M	IRQ Level				See Contiguous I/O Sample Tuple	TPCE_IR
	EEH	1	1	1	0	EH				IRQ Sharing Logic Active in Card Control & Status Register, Pulse and Level Mode Interrupts supported, recommended IRQ's IRQ14 for ATA Compatibility.	

Table 4-28: Sample No Link Tuple: 14H
Optional but recommended, for PC Card ATA Devices

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	14H	CISTPL_NO_LINK								No Link Control Tuple	Tuple Code
2: 1 Byte	00H									Link is 0 bytes	Link Length

This tuple prevents the CIS parser from attempting the implied long link to address 0 in common memory after parsing the first tuple chain on the card.

Table 4-29: Sample End of Tuple Chain: FFH
Required for PC Card ATA Devices as the last tuple on each chain which is not terminated with a link value of FFH.

Order: Size	Data	7	6	5	4	3	2	1	0	Description of Contents	CIS Function
1: 1 Byte	FFH	CISTPL_END								End of List Tuple	Tuple Code

©1999PCMCIA/JEIDA

4.7 Guideline for CIS Tuples for 3.3 or 3.3/5 volt Operation

4.7.1 Introduction

4.7.1.1 Purpose

This document is designed to provide implementation examples and further explanations of the PC Card standards in order to:

- enhance the interoperability of PC Card components, including card hardware and software, system hardware and software, and applications.
- facilitate the development of PC Card hardware and software by increasing the understanding of the standard by PC Card implementation community.

These guidelines are not requirements made by the PCMCIA/JEIDA standards organizations. Rather, they are implementation examples, suggestions and hints.

4.7.1.2 Scope

This document presents CIS examples for three distinct PC Card implementation options. Section 4.7.2 CIS for a PC Card with 3.3 volt Only Operation, provides a CIS example for a 16-bit PC Card that operates at 3.3 volts **V**CC only. Section 3 provides a CIS example for a 16-bit PC Card that can operate at either 3.3 or 5 volts. Section 4 provides a CIS example for a CardBus PC Card.

The CIS examples shown in this document are intended to describe only the tuples and tuple fields effected by the 3.3 or 3.3/5 volt operation characteristics of the card. The implementor should refer to the guideline for the specific card being implemented, for example, MODEM, ATA, or CardBus, for the general CIS guideline for the card and consider this guideline additional material dealing with 3.3 or 3.3/5 volt operation that may add tuples or define certain tuple fields.

This guidelines section is meant to be an illustration of how the ***PC Card Standard*** is used and not an additional source of such standards.

© 1999 PCMCIA/JEIDA

4.7.2 CIS for a PC Card with 3.3 volt Only Operation

4.7.2.1 CISTPL_DEVICE_OC

In this guideline, the CISTPL_DEVICE_OC has the VCC field set to one (1) in the Other Conditions Information Field and provides Device Info fields describing operation at 3.3 volts. If a CISTPL_DEVICE_A tuple exists, CISTPL_DEVICE_OA tuple shall also be present to describe the 3.3 volt operation of attribute memory.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1CH									CISTPL_DEVICE_OC
12	03H									Link
23	02H	Ext (0)	Rsvd (0)				VCC(01)		MWAIT (1)	Other conditions information field. In this example there is no extension and VCC is set to one indicating 3.3 volt operation.
34	D9H	Device Type (DH)				WPS (1)	Device Speed (1H)			Device Info field. In this example the device type is function specific (DH), WPS indicates the device is always writeable, and the device speed is shown as 250 ns.
45	FFH									End of device info

©1999PCMCIA/JEIDA

4.7.2.2 CISTPL_CONFIG

A CISTPL_CONFIG may exist to describe interface, power, timing, window, interrupt, and other information for 3.3 volt operation. This shall be followed by CISTPL_CFTABLE_ENTRY tuples for each configuration supported by the PC Card. In this example, two configuration entries are shown describing two different interface alternatives, both at 3.3 volts.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1AH									CISTPL_CONFIG
12	05H									Link
23	01H	TPCC_RFSZ (0)		TPCC_RMSZ (0)				TPCC_RASZ (1)		Size of fields. In this example, 1 mask is defined and the configuration register address is 2 byte in length.
34	01H	**RFU** (0)		Last Index (1)						Configuration Table Last Entry Number. In this example, the last configuration index is 1.
45	00H	Low order address								Bytes 5 and 6 provide the Configuration Option Register address. In this example that address is 200H.
56	02H	High order address								
67	07H	Config reg presence mask								Configuration register presence mask. In this example, three registers are declared: Configration Option, Configuration and Status, and Pin Replacement.

4.7.2.3 CISTPL_CFTABLE_ENTRY

A CISTPL_CFTABLE_ENTRY for each different configuration supported by the PC Card follows the CISTPL_CONFIG tuple. Following the example above, two configurations are described.

© 1999 PCMCIA/JEIDA

4.7.2.3.1 First CISTPL_CFTABLE_ENTRY

The first configuration describes a memory interface at 3.3 volts **Vcc**.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1BH									CISTPL_CFTABLE_ENTRY
12	0DH									Link
23	C0H	Int (1)	Def (1)	Conf Index (0)						Configuration Index byte. In this example, the configuration index number is 0, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
34	COH	Wt (1)	R (1)	WP (0)	BVD (0)	Int Type (0)				Interface description field. In this example the byte states that **WAIT#** and **READY** are active, Write Protect and BVD's are not used, and the interface type is memory.
45	21H	Mis (0)	Mem (01)		IRQ (0)	IO (0)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and a single 2-byte length memory space are described.
56	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
67	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 7 and 8 describe that the nominal **Vcc** is 3.3 volts.
78	1EH	Ext (0)	Extension (1EH/ 30B)							
89	35H	Ext (0)	Mantissa (0110)				Exponent (101)			Describes the minimum **Vcc** as 3.0 volts.
910	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 3.6 volts.
1011	3CH	Ext (0)	Extension (3CH/ 60B)							
1112	1EH	Ext (0)	Mantissa (0011)				Exponent (110)			Describes average current as 150ma.
1213	46H	Ext (0)	Mantissa (1000)				Exponent (110)			Describes peak current as 400ma.
1314	08H									Bytes 14 and 15 describe a 2K (8x256 bytes) memory window starting at address 0000.
1415	00H									

©1999PCMCIA/JEIDA

4.7.2.3.2 Second CISTPL_CFTABLE_ENTRY

The second configuration describes an I/O interface at 3.3 volts **Vcc**.

Byte	Data	7	6	5	4	3	2	1	0	Notes
0	1BH									CISTPL_CFTABLE_ENTRY
1	0CH									Link
2	C1H	Int (1)	Def (1)	Conf Index (1)						Configuration Index byte. In this example, the configuration index number is 1, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
3	41H	Wt (0)	R (1)	WP (0)	BVD (0)	Int Type (1)				Interface description field. In this example the byte states that **READY** is active, **WAIT#**, Write Protect and BVD's are not used, and the interface type is I/O.
4	09H	Mis (0)	Mem (00)		IRQ (0)	IO (1)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and I/O space are described.
5	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
6	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 7 and 8 describe that the nominal **Vcc** is 3.3 volts.
7	1EH	Ext (0)	Extension (1EH/ 30B)							
8	35H	Ext (0)	Mantissa (0110)				Exponent (101)			Describes the minimum **Vcc** as 3.0 volts.
9	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 3.6 volts.
10	3CH	Ext (0)	Extension (3CH/ 60B)							
11	1EH	Ext (0)	Mantissa (0011)				Exponent (110)			Describes average current as 150ma.
12	46H	Ext (0)	Mantissa (1000)				Exponent (110)			Describes peak current as 400ma.
13	64H	Rng (0)	16B (1)	8B (1)	IO address lines (00100)					Describes an I/O window containing 16 contiguous registers accessable in either 8 or 16 bit accesses.

Note: Since the first table entry described only a memory window and the second only an I/O window, Default had to be set in the second entry to eliminate the memory window described in the first entry. If both entries had been memory windows or both entries had been I/O windows, the Default bit could have been negated in the second entry and the power description not repeated.

© 1999 PCMCIA/JEIDA

4.7.3 CIS for a PC Card with 3.3 or 5 volt Operation

4.7.3.1 CISTPL_DEVICE

As noted previous, the CISTPL_DEVICE tuple is required to be the first tuple in the CIS. The CISTPL_DEVICE tuple indicates characteristics for 5 volt operation and a CISTPL_DEVICE_OC tuple shall exist to indicate 3.3 volt operation and characteristics.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	01H									CISTPL_DEVICE
12	02H									Link
23	DAH	Device Type (DH)				WPS (1)	Device Speed (2H)			Device Info field. In this example the device type is function specific (DH), WPS indicates the device is always writable, and the device speed is shown as 200ns.
34	08H	Number of address units - 1 (01H)					Size Code (0H)			Device Size (Two 512 byte units or 1 K byte)

A CISTPL_DEVICE_A tuple may also be present in the CIS. The CISTPL_DEVICE_A tuple also describes 5 volt operation and requires a CISTPL_DEVICE_OA tuple be present to describe 3.3 volt operation.

4.7.3.2 CISTPL_DEVICE_OC

In this example, the CISTPL_DEVICE_OC has the VCC field set to one (1) in the Other Conditions Information Field and provides Device Info fields describing operation at 3.3 volts.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1CH									CISTPL_DEVICE_OC
12	03H									Link
23	02H	Ext (0)	Rsvd (0)				VCC (01)		MWAIT (1)	Other conditions information field. In this example there is no extension, VCC is set to one indicating 3.3 volt operation.
34	D9H	Device Type (DH)				WPS (1)	Device Speed (1H)			Device Info field. In this example the device type is function specific (DH), WPS indicates the device is always writable, and the device speed is shown as 250ns.
45	FFH									End of device info

If a CISTPL_DEVICE_A tuple exists, a CISTPL_DEVICE_OA tuple shall also be present describing 3.3 volt operation of attribute memory.

©1999PCMCIA/JEIDA

Note: When the Device Info field in both the CISTPL_DEVICE and CISTPL_DEVICE_OC tuples is non-zero the host is made aware that the card is capable of operating at either 3.3 or 5 volts. In addition, the host knows that the common memory cycle time is 200ns when operating at 5 volts and 250ns when operating at 3.3 volts.

© 1999 PCMCIA/JEIDA

4.7.3.3 CISTPL_CONFIG

A CISTPL_CONFIG tuple may exist to describe the interface, power, window, and other information for 3.3 and 5 volt operation. The CISTPL_CONFIG tuple is followed by CISTPL_CFTABLE_ENTRY tuples for each configuration supported by the PC Card. In this example, four configuration entries are described, one for each of two interface alternatives at each of the two voltage alternatives.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1AH									CISTPL_CONFIG
12	05H									Link
23	01H	TPCC_RFSZ (0)		TPCC_RMSZ (0)				TPCC_RASZ (1)		Size of fields. In this example 1 mask is defined and the configuration register address is 2 byte in length.
34	11H	**RFU** (0)		Last Index (11H)						Configuration Table Last Entry Number. In this example the last configuration index is 11H.
45	00H	Low order address								Bytes 5 and 6 provide the Configuration Option Register address. In this example that address is 200H.
56	02H	High order address								
67	07H	Config reg presence mask								Configuration register presence mask. In this example three registers are declared:, Configuration Option, Configuration and Status, and Pin Replacement.

4.7.3.4 CISTPL_CFTABLE_ENTRY

A CISTPL_CFTABLE_ENTRY tuple for each configuration supported by the PC Card follows the CISTPL_CONFIG tuple. Following the example above, four configurations are described.

©1999PCMCIA/JEIDA

4.7.3.4.1 First CISTPL_CFTABLE_ENTRY

The first configuration describes a Memory interface at 3.3 volts **Vcc**.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1BH									CISTPL_CFTABLE_ENTRY
12	0DH									Link
23	C0H	Int (1)	Def (1)	Conf Index (0)						Configuration Index byte. In this example, the configuration index number is 0, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
34	COH	Wt (1)	R (1)	WP (0)	BVD (0)	Int Type (0)				Interface description field. In this example the byte states that **WAIT#** and **READY** are active, Write Protect and BVD's are not used, and the interface type is memory.
45	21H	Mis (0)	Mem (01)		IRQ (0)	IO (0)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and a single 2-byte length memory space are described.
56	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
67	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 7 and 8 describe that the nominal **Vcc** is 3.3 volts.
78	1EH	Ext (0)	Extension (1EH/ 30B)							
89	35H	Ext (0)	Mantissa (0110)				Exponent (101)			Describes the minimum **Vcc** as 3.0 volts.
910	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 3.6 volts.
1011	3CH	Ext (0)	Extension (3CH/ 60B)							
1112	1EH	Ext (0)	Mantissa (0011)				Exponent (110)			Describes average current as 150ma.
1213	46H	Ext (0)	Mantissa (1000)				Exponent (110)			Describes peak current as 400ma.
1314	08H									Bytes 14 and 15 describe a 2K memory window starting at address 0000.
1415	00H									

© 1999 PCMCIA/JEIDA

4.7.3.4.2 Second CISTPL_CFTABLE_ENTRY

The second configuration describes an I/O interface at 3.3 volts **Vcc**.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1BH									CISTPL_CFTABLE_ENTRY
12	0CH									Link
23	C1H	Int (1)	Def (1)	Conf Index (1)						Configuration Index byte. In this example, the configuration index number is 1, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
34	41H	Wt (0)	R (1)	WP (0)	BVD (0)	Int Type (1)				Interface description field. In this example the byte states that **READY** is active, **WAIT#**, Write Protect and BVD's are not used, and the interface type is I/O.
45	09H	Mis (0)	Mem (00)		IRQ (0)	IO (1)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and I/O space are described.
56	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
67	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 7 and 8 describe that the nominal **Vcc** is 3.3 volts.
78	1EH	Ext (0)	Extension (1EH/ 30B)							
89	35H	Ext (0)	Mantissa (0110)				Exponent (101)			Describes the minimum **Vcc** as 3.0 volts.
910	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 3.6 volts.
1011	3CH	Ext (0)	Extension (3CH/ 60B)							
1112	1EH	Ext (0)	Mantissa (0011)				Exponent (110)			Describes average current as 150ma.
1213	46H	Ext (0)	Mantissa (1000)				Exponent (110)			Describes peak current as 400ma.
1314	64H	Rng (0)	16B (1)	8B (1)	IO address lines (00100)					Describes an I/O window containing 16 contiguous registers accessable in either 8 or 16 bit accesses.

©1999PCMCIA/JEIDA

4.7.3.4.3 Third CISTPL_CFTABLE_ENTRY

The third configuration describes a memory interface at 5 volt VCC.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1BH									CISTPL_CFTABLE_ENTRY
12	0DH									Link
23	C2H	Int (1)	Def (1)	Conf Index (10H)						Configuration Index byte. In this example, the configuration index number is 10H, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
34	COH	Wt (1)	R (1)	WP (0)	BVD (0)	Int Type (0)				Interface description field. In this example the byte states that **WAIT#** and **READY** are active, Write Protect and BVD's are not used, and the interface type is memory.
45	21H	Mis (0)	Mem (01)		IRQ (0)	IO (0)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and a single 2-byte length memory space are described.
56	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
67	55H	Ext (0)	Mantissa (1010)				Exponent (101)			Describes that the nominal **Vcc** is 5 volts.
78	C5H	Ext (1)	Mantissa (1000)				Exponent (101)			Bytes 8 and 9 describe the minimum **Vcc** as 4.75 volts.
89	4BH	Ext (0)	Extension (4BH/ 75B)							
910	D5H	Ext (1)	Mantissa (1010)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 5.25 volts.
1011	19H	Ext (0)	Extension (19H/ 25B)							
1112	0EH	Ext (0)	Mantissa (0001)				Exponent (110)			Describes average current as 120ma.
1213	3EH	Ext (0)	Mantissa (0111)				Exponent (110)			Describes peak current as 350ma.
1314	08H									Bytes 14 and 15 describe a 2K memory window starting at address 0000.
1415	00H									

© 1999 PCMCIA/JEIDA

4.7.3.4.4 Fourth CISTPL_CFTABLE_ENTRY

An I/O interface at 5 volts VCC.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1BH									CISTPL_CFTABLE_ENTRY
12	0CH									Link
23	C3H	Int (1)	Def (1)	Conf Index (11H)						Configuration Index byte. In this example, the configuration index number is 11H, the interface flag indicates that the interface will be described, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
34	41H	Wt (0)	R (1)	WP (0)	BVD (0)	Int Type (1)				Interface description field. In this example the byte states that **READY** is active, **WAIT#**, Write Protect and BVD's are not used, and the interface type is I/O.
45	09H	Mis (0)	Mem (00)		IRQ (0)	IO (1)	Tim (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and I/O space are described.
56	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
67	55H	Ext (0)	Mantissa (1010)				Exponent (101)			Describes that the nominal **Vcc** is 5 volts.
78	C5H	Ext (1)	Mantissa (1000)				Exponent (101)			Bytes 8 and 9 describe the minimum **Vcc** as 4.75 volts.
89	4BH	Ext (0)	Extension (4BH/ 75B)							
910	D5H	Ext (1)	Mantissa (1010)				Exponent (101)			Bytes 10 and 11 describe that the maximum **Vcc** is 5.25 volts.
1011	19H	Ext (0)	Extension (19H/ 25B)							
1112	0EH	Ext (0)	Mantissa (0001)				Exponent (110)			Describes average current as 120ma.
1213	3EH	Ext (0)	Mantissa (0111)				Exponent (110)			Describes peak current as 350ma.
1314	64H	Rng (0)	16B (1)	8B (1)	IO address lines (00100)					Describes an I/O window containing 16 contiguous registers accessable in either 8 or 16 bit accesses.

Note: By having two entries for each window, one at each VCC level, a host capable of operating the card at either 3.3 or 5 volts can choose the better power consumption alternative, 495 milliwatts average at 3.3 volts vs 600 milliwatts average at 5 volts, or the better performance alternative, 200ns cycle time at 5 volts vs 250 ns cycle time at 3.3 volts.

©1999PCMCIA/JEIDA

4.7.4 CIS for a CardBus PC Card

Since a CardBus PC Card only operates at 3.3 volts **V**CC, a CISTPL_DEVICE tuple is not required. If memory space is used, a CISTPL_DEVICE_OC tuple is required. A CISTPL_CONFIG_CB tuple and a CISTPL_CFTABLE_ENTRY_CB tuple are required for each function on the card.

4.7.4.1 CISTPL_DEVICE_OC

The CISTPL_DEVICE_OC has the VCC field set to one (1) in the Other Conditions Information Field and provides Device Info fields describing operation at 3.3 volts.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	1CH									CISTPL_DEVICE_OC
12	03H									Link
23	02H	Ext (0)	Rsvd (0)				**Vcc** (01)		MWAIT(0)	Other conditions information field. In this example there is no extension, VCC is set to one indicating 3.3 volt operation.
34	D8H	Device Type (DH)				WPS (1)	Address Space Indicator (1H)			Device Info field. In this example the device type is function specific (DH), WPS indicates the device is always writable.
45	FFH									End of device info

4.7.4.2 CISTPL_CONFIG_CB

A CISTPL_CONFIG_CB descibes the interface, power, timing, window, interrupt, and other information for 3.3 volt operation. The CISTPL_CONFIG_CB tuple shall be followed by CISTPL_CFTABLE_ENTRY_CB tuples for each configuration supported by the PC Card. In this example, one configuration entry is shown describing operation at 3.3 volts.

Byte	Data	7	6	5	4	3	2	1	0	Notes
01	04H									CISTPL_CONFIG_CB
12	06H									Link
23	03H	TPCC_RFSZ (0)		Reserved, must be zero				TPCC_ADDR (3)		Size of fields. For CardBus PC Cards the four status registers are always present.
34	00H	**RFU** (0)		Last Index (0)						Configuration Table Last Entry Number. In this example the last configuration index is 0.
45	01H	Address Space Offset LSB (0)				Rsvd (0)	Address Space Indicator(1)			Register Address. Reside in Base Memory Area 1 at offset 100H.
56	01H	Address Space Offset 2 (01H)								
67	00H	Address Space Offset 3 (00H)								
78	00H	Address Space Offset MSB (00H)								

© 1999 PCMCIA/JEIDA

4.7.4.3 CISTPL_CFTABLE_ENTRY_CB

Following the CISTPL_CONFIG_CB will be a CISTPL_CFTABLE_ENTRY_CB for each different configuration alternative that the function can provide. Following the example from above, one configuration is described.

Byte	Data	7	6	5	4	3	2	1	0	Notes
0	05H									CISTPL_CFTABLE_ENTRY_CB
1	0CH									Link
2	40H	**RFU** (0)	Def (1)	Conf Index (0)						Configuration Index byte. In this example, the configuration index number is 0, and the default flag is set indicating that defined values replace any previously defined values but may be used in following entries.
3	21H	Mis (0)	**RFU** (0)	Mem (1)	IRQ (0)	IO (0)	**RFU** (0)	Pwr (01)		Feature selection field. In this example **Vcc** power, and a Memory Space descriptor are present.
4	37H	**RFU** (0)	Prdn I (0)	Peak I (1)	Avg I (1)	Stat I (0)	Max V (1)	Min V (1)	Nom V (1)	Power description parameter selection field. In this example Nominal V, Minimum V, Maximum V, Average I, and Peak I will be described.
5	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 6 and 7 describe that the nominal **Vcc** is 3.3 volts.
6	1EH	Ext (0)	Extension (1EH/ 30B)							
7	35H	Ext (0)	Mantissa (0110)				Exponent (101)			Describes the minimum **Vcc** as 3.0 volts.
8	B5H	Ext (1)	Mantissa (0110)				Exponent (101)			Bytes 9 and 10 describe that the maximum **Vcc** is 3.6 volts.
9	3CH	Ext (0)	Extension (3CH/ 60B)							
10	1EH	Ext (0)	Mantissa (0011)				Exponent (110)			Describes average current as 150ma.
11	46H	Ext (0)	Mantissa (1000)				Exponent (110)			Describes peak current as 400ma.
12	04H	Memory Base Address Register (04H)							**RFU** (0)	Base Address Register 4 describes the memory.

©1999PCMCIA/JEIDA