PCMCIA Training Manual

Volume I:

"An Introduction to Flash Memory PC Cards"

Version 1.0

A D V A N C E D   M I C R O   D E V I C E S

Trademarks

Table of Contents

Introduction

Personal computer end users are moving to more compact and portable systems. This move to portability has created a need for systems with light weight, small size, and extended battery life. In addition, end users are requiring a new level of ruggedness, reliability, and data compatibility across different host system platforms. Advanced Micro Devices has always been a conscientious supplier to computer system OEMs, and AMD feels that devotion to portable computing is the required response to its customer's needs.

AMD supplies a complete line of products to computer system OEMs including Flash and UV EPROM Memory products, CISC Microprocessors (e.g. Am386 ™), RISC Microprocessors (e.g. the 29K family), peripheral control devices such as Small Computer System Interface (SCSI) devices, Ethernet networking products, and Programmable Logic Devices. In addition, many products within these broad product lines are being designed with the needs of the portable computer end users in mind: 5.0 volt-only Flash memory, 3.3 volt EPROMs, OTPROM, SCSI, and Am386s.

AMD sees Flash memory as an important technology in portable computing. In 1992, AMD and Fujitsu announced that they were forming a joint venture to manufacture and market advanced Flash and EPROM Memory products. In the joint venture, AMD and Fujitsu will invest equally in a $700 million manufacturing facility to produce memory devices at sub 0.5 micron geometry by the end of 1994. Also in 1992, AMD formed a partnership with Du Pont Connector Systems to manufacture and market Flash memory cards. These cards meet the Personal Computer Memory Card International Association (PCMCIA) standard.

The PCMCIA standard Memory Cards and Input/Output (I/O) Cards represent the state of the art in portable computer peripherals. These cards are referred to by many names including PCMCIA Cards, PC Memory Cards, PC I/O Cards, and PC Cards. Throughout the text, the authors use PC Cards to mean any Memory or I/O Card that meets the minimum compliance requirements of the PCMCIA standard. Please note that *italics* are used throughout the manual when defining a word or term. This is done to assist the user with the terminology.

PC Cards are designed for systems that require portability and high reliability. In addition, the PCMCIA standard addresses the need for PC Cards to be accessed by different host system platforms. The standard was developed to eliminate potential damage to the system or PC Card even if a PC Card was put into a PCMCIA socket that doesn't have the capability to interact with the card.

On one hand, end users need to understand the basics of PCMCIA cards in order to be informed of the capabilities of future generations of portable computers. On the other hand, system designers need to keep abreast of the latest developments in PC Cards in order to incorporate features that meet the requirements of an ever changing user group. Like the introduction of the local area network (LAN) and the mouse, PC cards are a new member of the family of elements that make up the personal computing environment. In fact, the PC Card is one of the most important elements of a portable computer. In particular, PC Cards offer the portability, low power consumption, and ruggedness that only solid state design can provide. PC Cards are one of the enabling technologies for a truly portable computer.

AMD's Flash memory PC cards are based on both our family of 5.0 volt read/12.0 volt write, and 5.0 volt-only Flash memory devices. AMD's Flash memory cards with Embedded Algorithms provide a minimum 100,000 write cycles and unlimited read cycles. In addition, AMD's 5.0 volt-only Flash memory cards provide sector erase capability so that entire device need not be "bulk erased". Chapter 1, "AMD's Flash memory PC Cards", gives an overview of the Flash memory cards that AMD offers.

AMD also has a cadre of third party suppliers to provide system-level software and device drivers for our Flash memory cards. Companies that are members of AMD's FusionFlash Team, and the support services that they provide are listed in Chapter 2, "AMD's FusionFlash Partners".

Objective of this Manual

The intent of this manual is to:

- explain the PCMCIA standard
- describe PCMCIA hardware, firmware, and software
- define the terminology associated with PCMCIA
- give the reader familiarity with PC Cards

Thus, the manual consolidates and explains much of the literature regarding PC Cards and the PCMCIA standard. The manual will serve the following three roles:

1. A training manual targeted to a broad base of readers, with various levels of technical background, for the purpose of gaining an understanding of PC Cards, the PCMCIA standard, and how AMD's Flash memory PC Cards fill a need in portable computing.

2. A PC Card user's guide that will provide a better understanding of the way PC Cards work in systems.

3. A technical reference for application engineers and system designers.

Since this manual addresses such a broad audience and range of topics, a roadmap to the manual and some suggestions of how to use the manual for different purposes have been provided. The "Roadmap to the Manual" section is a quick guide to show a given reader where to start in the manual given his or her background and area of interest.

The "How to Use the Manual" section shows how different parts of the manual can be omitted by different readers. In addition, the authors have included a list of "Related Documents" for readers that would like more information or details about topics that are not covered in depth in this manual.

Roadmap to the Manual

This Training Manual is intended to be a multiple volume document. Volume I of this Training Manual approaches the topic of PC Cards and the PCMCIA standard from the ground up. The material in Volume I is organized into three main parts and a set of appendices. Volume I is intended for the users of AMD's Flash memory Cards and it is concentrated in the PC Memory Card area. Volume II, to be published in the near future, will give many of the technical details about PC Cards and the PCMCIA standard and is intended for the person who is familiar with the basics of PCMCIA. The content of Volume II are still in the definition stage and some of the Volume II topics is listed in the "Possible Topics of PCMCIA Training Manual - Volume II" diagram.

Roadmap to the PCMCIA Training Manual - Volume I

AREA OF INTEREST:

1- AMD's PCMCIA Flash Memory Cards 2- AMD's Fusion Partners for Flash Memory Cards 3- Needs of a Portable Computer User	**PART I** Chapters 1 2 3
4- Introduction of PCMCIA & Overview of the PCMCIA Standard 5- PCMCIA Memory Card & I/O Card Interface 6- PCMCIA Hardware and Socket Services	**PART II** Chapters 4 5 6
7- PCMCIA Software / Metaformat / tuple, Card Services / Memory Technology Drivers Flash File Systems 8- How to perform Erase / Write / File access on an AMD Flash Memory Card	**PART III** Chapters 7 8
A- Solid State vs. Rotating Magnetic Media Mass Storage B- IBM PC Fundamentals C- PC Card Pin Assignments	**Appendix** A B C

Possible Topics for the PCMCIA Training Manual - Volume II

- Hardware:
  Signal Description, Interface Controllers
- Socket Services
- Software:
  Metaformat, Tuples, Card Information Structure, Card Services, Memory Technology Drivers
- Flash File Systems
- X86 Microprocessor Fundamentals
- Exchangeable Card Architecture (ExCA™)
- eXecute-In-Place (XIP)
- PCMCIA Working Groups
- Auto-Indexing Mass Storage (AIMS)
- Advanced Technology Attachment (ATA)

Appendices

This manual also includes three appendices containing topics considered supplementary to the core sections of the manual.

- Appendix A, "Solid State versus Rotating Mass Storage"
  A comparison of solid state memory versus rotating magnetic media as a mass storage area for the PC. The comparison provides distinct advantages and disadvantages of each method of mass storage.
- Appendix B, "The IBM Compatible Personal Computer"
  A brief history of the IBM PC and describes the major components of an IBM Compatible PC and how they interact.
- Appendix C, "PCMCIA PC Card Pin Assignments"
  A table of the PCMCIA Memory cards and I/O cards pin assignments.

How to Use the Manual

The manual is designed to be used by people with different levels of interest in PCMCIA, and different backgrounds about PC Cards and the PCMCIA standards. In addition, the manual is designed as a tutorial as well as a reference text. A person with minimal understanding of the personal computer should be able to start at Chapter 1 and by the end of Volume I have an understanding of the general trends in portable computing, the basic operation of a personal computer, and an intermediate level of understanding about PCMCIA cards and specifications, and AMD's PCMCIA products and support services.

For the person who is unfamiliar with the inner workings of a PC or needs a refresher on the PC, Appendix B of Volume I is an excellent overview.

For the user of the manual that wants an in depth treatment of PCMCIA hardware and software topics and is already familiar with the basics of PCMCIA, Volume II of the manual is appropriate.

Finally, if the user of the manual is interested in the details of a special related topic, the Appendices and Related Documents give in depth treatment of specific issues.

Related Documents

- PCMCIA PC Card Standard Release 1.0 and 2.0
- PCMCIA Card Services Interface Specification Release 2.0
- PCMCIA Socket Services Interface Specification Release 1.01 and 2.0
- Using MS-DOS
- Que's Computer User's Dictionary
- Electronic Buyers' news
- EE Times
- Electronic News
- Memory Card Systems and Design
- IC Card systems and design

Part I - AMD's Flash memory PC Cards, Third Party Support, and Portable Computing

Part I is an explanation of the role that AMD plays in the world of PCMCIA and the PCMCIA products that AMD offers. It also discusses the needs of the portable computer user.

Chapter 1, "AMD's Flash memory PC Cards", gives some of the current and expected future features of our Flash memory PC Cards.

Chapter 2, "AMD's FusionFlash Partners", describes the types of third party support that our partner companies offer for users of AMD Flash memory PC Cards.

Chapter 3, "The Portable Computer User", identifies the changing needs of the portable computer user in comparison to the desktop user and explains how these changing needs translate into new technology requirements.

Chapter 1. AMD's Flash memory PC Cards

In July, 1992, AMD announced their participation in PCMCIA Flash memory card products through a manufacturing agreement with Du Pont. Two families of the PCMCIA standard 1.0 compliant Flash memory card were introduced in the fourth quarter of 1992. Both families are available in 1, 2 and 4 Megabyte capacities. The first family of cards (the AmC00xFLKA) are software compatible with Intel's 10K cycle "Series 1" cards. It is intended that this version of AMD's cards will be compatible with any socket interacting with an Intel "Series 1" Flash memory card. The second family (the AmC00xAFLKA) differs from the first only in the software algorithms required to write to the card. Different software algorithms are required to support the 100K cycle Embedded Algorithm devices used in these cards. The "A" family of cards provide an order of magnitude longer operating life than Intel's "Series 1" Cards. The "A" family is identified by the letter "A" in the part number "AmC00xAFLKA".

A third family of Flash memory cards is in the planning stage. This family will use AMD's 5.0 volt-only Flash memory devices. The 5.0 volt-only Flash memory cards will comply with the PCMCIA standard and they will require a different set of programming algorithms. The "B" family of cards also support the 100K cycle endurance using the Embedded Algorithm devices.

Available 4Q92: 12.0 Volt Write / 5.0 Volt Read

Family	Part Number	Endurance Cycles (min.)	Access Time	Program / Erase (1)	Supply Voltage (2)	Programming Voltage (2)
1st	AmC001FLKA	10 K	250 ns	Software	5 .0 V	12.0 V
1st	AmC002FLKA	10 K	250 ns	Software	5 .0 V	12.0 V
1st	AmC004FLKA	10 K	250 ns	Software	5 .0 V	12.0 V
2nd	AmC001AFLKA	**100 K**	250 ns	Embedded	5 .0 V	12.0 V
2nd	AmC002AFLKA	**100 K**	250 ns	Embedded	5 .0 V	12.0 V
2nd	AmC004AFLKA	**100 K**	250 ns	Embedded	5 .0 V	12.0 V

Available 2Q93: 5.0 Volt-Only Write / Read

Family	Part Number	Endurance Cycles (min.)	Access Time	Program / Erase (1)	Supply Voltage (2)	Programming Voltage (2)
3rd	AmC001BFLKA	**100 K**	150 ns	Embedded	5.0 V	5.0 V
3rd	AmC002BFLKA	**100 K**	150 ns	Embedded	5.0 V	5.0 V

Notes:
(1) Program / Erase:
Software = Intel compatible Software controlled Flashrite ™ and Flasherase ™ Algorithms
Embedded = Embedded Program ™ and Embedded Erase ™ Algorithms
(2) All supply and programming voltages are +/- 5%

Chapter 2. AMD's FusionFlash Partners

AMD has worked closely with both hardware and software vendors to ensure that our Flash memory cards are compatible with many systems and platforms available today. The following is a partial list of vendors who have worked with AMD.

A Memory Card Reader / Programmer is also know as a "Memory Card Drive". It is similar to a floppy disk drive in a personal computer. A Memory Card Drive could be used to erase and format a Flash memory card. It is also used to program data files into the card. A PC Card Drive has the capability to access both Memory PC Cards and I/O PC Cards. An example of the usage of an AMD Flash memory card in the Databook Card Drive is in Chapter 8 of this Training Manual.

MEMORY CARD READER / PROGRAMMER

Company	Address / Phone / Fax
Adtron Corp.	128 West Boxelder Place Suite 102, Chandler, AZ 85224 Phone: (602) 926-9324 Fax: (602) 926-9359
Databook Inc.	Tower Building, Terrace Hill, Ithaca, New York 14850 Phone: (607) 277-4817 Fax: (607) 273-8803
Data IO	10525 Willows Road N.E. PO Box 97046 Redmond, WA 98073-9746 Phone : (206) 881-6444 Fax: (206) 882-1043
SCM Microsystem GmbH	Berta von Suttnerweg 1 D-8033 Martinsried, West Germany Phone: (089) 856-1363 Fax: (089) 856-1412 or Frauenhofer StraBe 11A D-8033 Martinsried, West Germany Phone: (089) 859-8702 Fax: (089) 859-5806

PCMCIA compliant system and Flash memory cards require software support to perform the erase, format, and programming functions. Details of the various types of software support are in Chapter 7 of this Training Manual.

PCMCIA SYSTEM-LEVEL AND MEMORY CARD SOFTWARE SUPPORT

Company	Address / Phone / Fax	Types of Support
Award Software Inc.	130 Knowles Drive, Los Gatos, CA 95030-1832 Phone: (408) 370-7979 Fax: (408) 379-3399	Card Services, Socket Services, MTDs, device drivers
Microsoft Corp.	One Microsoft Way, Redmond, WA 98052 Phone: (800) 426-9400	Flash File System
Phoenix Technologies Ltd.	846 University Ave, Norwood Mass 02062 Phone: (617) 551-4175 Fax: (617) 551-3743	Card Services, Socket Services, MTDs, device drivers

MEMORY CARD MECHANICAL HARDWARE

Company	Address / Phone	Types of Support
Du Pont Connectors System	14 T.W. Alexander Drive, P.O. Box 13999 Research Triangle Park, NC 27709 Phone: (800) 237-2374	Card Connectors, Card Ejectors, Card Manufacturing

Chapter 3. The Portable Computer User

The portable segment is the fastest growing in the computer market. Success in the portable market segment will be based on the computer system *Original Equipment Manufacturers'* (OEM) or the components manufacturers' ability to respond to the changing demands of the portable computer user. Capturing a proportionally large share of the market and profits from this exciting growth industry will be based on a company's ability to recognize and address the changing demands, in terms of functionality, that *end users* require in portables. A functional perspective of the way that mobile workers will use products provides a good model to identify the required product features of portable computers in the future. Portable computer end users will demand a different set of performance criteria than today's desktop users.

The majority of portables will be used in a mobile environment by "desk-less" and "lap-less" workers. Figure 3.1 illustrates the various form factors of a personal computer.

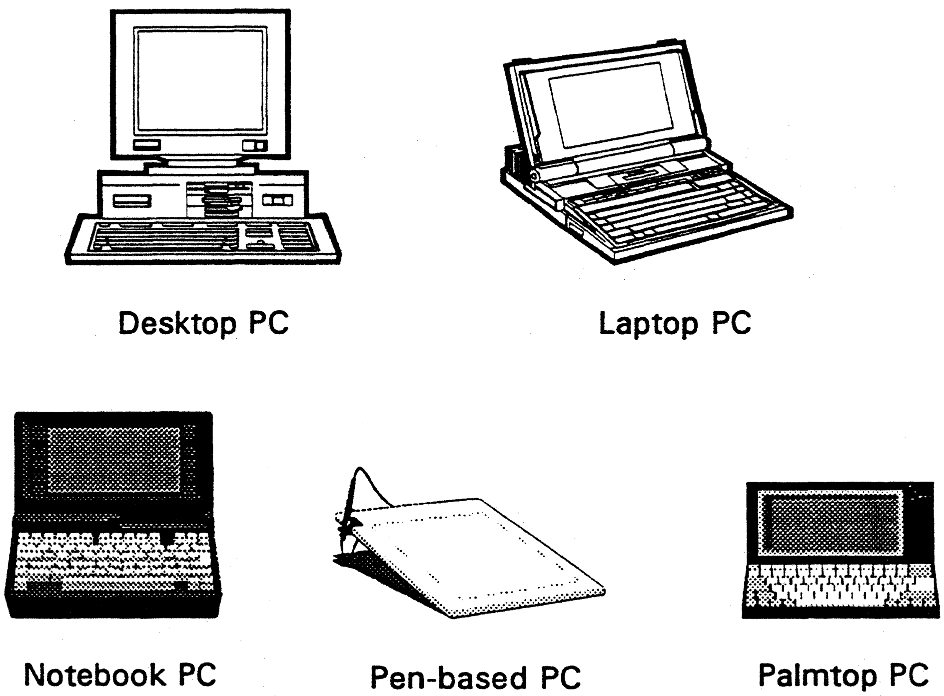

Figure 3.1 Personal Computer Form Factors

Mobile workers demand *Personal Information Devices* (PIDs) that:

- Are durable and rugged like electronic calculators. The time constrained mobile workers of tomorrow are likely to toss their PIDs onto the passenger seat of their truck, in the case of a delivery driver, or inadvertently drop the PID while checking factory inventories, in the case of an inventory control clerk.

- Have a useful battery life longer than a single working day without recharging the batteries in the middle of a client call or work shift.

- Are light weight and easy to handle. For example, the PIDs should not make your "arm grow longer" as it is carried through the airport. PIDs should be easily held while at work, unlike the Hertz return worker that carries a five pound battery pack attached to his or her belt.

The way computer OEMs and component manufactures address the requirements of portable computer users will be important to the development and growth of new technologies in the industry. One area that appears poised for radical change, and a topic of this manual, is the memory architecture of PCs. This is fueled by the increasing capabilities of integrated processors, low-power components, and Flash memory technologies focused on the PID segment. Figure 3.2, "Emerging Memory Architecture", shows one possible memory configuration for future PIDs in the hand-held and pen tablet form factors.

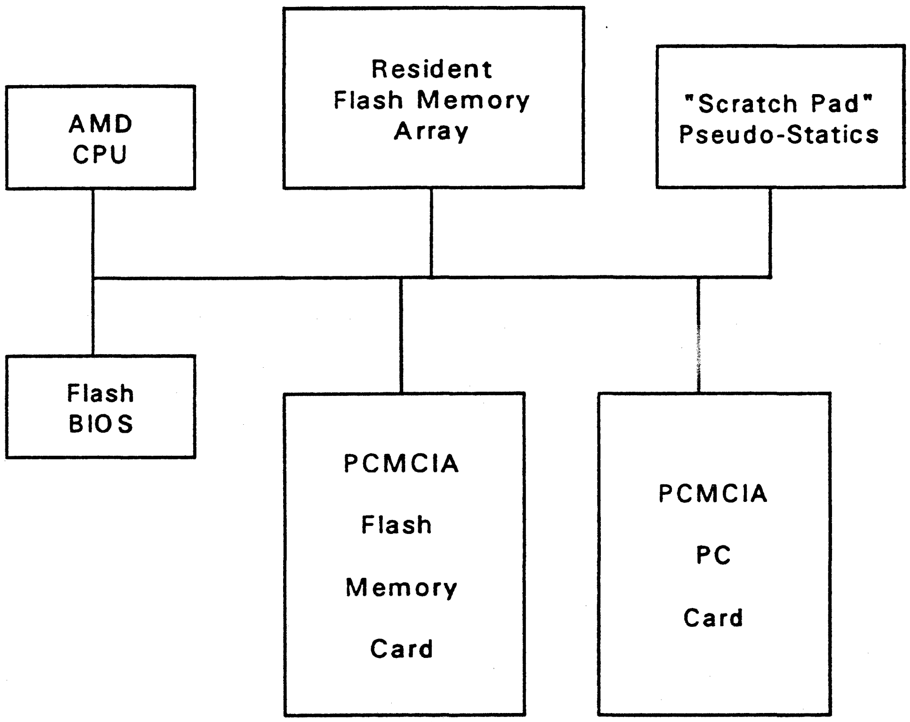

Figure 3.2 Emerging Memory Architecture

Figure 3.2 leads to two points:

1. *Resident Flash Arrays* in the smaller form factor machines is expected to displace DRAM content that was strictly used to store executable code. For example, a system might be built with 4 to 8 Megabytes (Mbytes) of Flash on the motherboard to store DOS, Windows, and some Personal Information Management software. Additional system memory is required for remaining traditional usage.

2. There will be a blurring in the distinction between main memory and mass storage. For example, a Flash memory PC Card might function as both a mass storage device and main memory when the system maps the PC Card contents into the main memory address space for direct execution of code.

Part II - Overview of the PCMCIA Standard and PC Cards

Part II introduces the concept of the PCMCIA hardware and software layers, and introduces the terminology of PCMCIA.

Chapter 4, "Introduction to PCMCIA", gives the history of PCMCIA, why it was established, and how it has evolved over the years. This chapter also introduces the PCMCIA hardware and software layers, and gives a diagram of the layers that is used throughout the remainder of the manual.

Chapter 5, "Understanding PCMCIA Cards and Connectors", is an introduction to the mechanical issues of PC Cards and sockets including a description of the physical interface, the dimensions for different card Types, the features available on different PC Card versions, and the environment in which cards may be used.

Chapter 6, "Understanding PCMCIA Hardware and Firmware", is an introduction to PCMCIA hardware from an electronic standpoint. This chapter also discusses Socket Services, the software interface to the PCMCIA hardware. Socket Services is expected to be provided as a BIOS extension and is the lowest layer of the PCMCIA software.

Chapter 4. Introduction to PCMCIA

4.1 The History of PCMCIA

The earliest Memory Cards were not actually targeted for portable computers. In fact, the development of the earliest card systems were for industrial PCs where the operating environment proved disk technology to be unreliable.

In June 1989, The Personal Computer Memory Card International Association (PCMCIA) was formed by a handful of companies that included Du Pont Electronics, Fujitsu, and Poqet Computer. In November 1990, the PCMCIA Standard, Release 1.0 was published. By this time eighty eight companies, including AMD, had become members of PCMCIA. The original intent of the standard was to define an interface for memory cards to be used for low power consumption, light weight, and reliable memory storage. The original Memory PC Cards used SRAM and battery backup on the card, OTPROM, Mask-ROM, and Flash memory. Thus, PCMCIA Standard Release 1.0 was defined only for the 68-pin PC Memory Cards. The 68-pin connector was originally defined by the Japanese Electronics Industry Association (JEIDA) to overcome various proprietary and incompatible IC-cards marketed in Japan. PCMCIA worked closely with JEIDA to ensure that the PCMCIA standard and the JEIDA standard were fully compatible with each other.

As the portable computer industry continued to grow, manufacturers that provided I/O products such as modems and network adapters became interested in access to portable computers. However, portable computers were so small that they typically had no room for internal expansion boards like desktop PCs. In addition, no portable bus or connector standard existed that was suitable for interchanging peripheral cards in mobile computers.

In September 1991, PCMCIA incorporated an I/O card interface specification into the PCMCIA Standard Release 2.0. Some of the components of the interface specification are Socket Services 1.01 and Metaformat additions. *Metaformat* is a PCMCIA term that describes the contents, format, and PC Card related information that are stored in the PC Cards. One important note, however, is that in order to maintain compliance to the original PCMCIA Standard Release 1.0, all PCMCIA compliant systems and PC Cards are required to power up in the PCMCIA Standard Release 1.0 configuration. Details about a PC Card powering up in this configuration are contained in Chapter 6, "PCMCIA Hardware and Firmware". Should the *host system* determine that the PC Card installed supports Revision 2.0, then the system will configure itself and the PC Card appropriately.

In July 1992, two more interface specifications, the PCMCIA - *Advanced Technology Attachment* (ATA) Mass Storage 1.0 and the *Auto Index Mass Storage* (AIMS) 1.0 specification were approved for PC Cards. The PCMCIA-ATA specification is an I/O implementation of an ATA mass storage protocol peripheral on a PC Card. The AIMS specification defines a standard for storing large data sets such as image and multimedia data files. An AIMS PC Card can be configured to operate as a PC I/O Card or a PC Memory Card. In September 1992, two software interface specifications, Card Services 2.0 and Socket Services 2.0 have been approved by the members of PCMCIA. Those two software specifications are expected to provide a mean to allow PCMCIA compliant systems for a specific operation system to communicate with each other. More details of Card Services and Socket Services will be provided in chapters 6 and 7 of this manual. By September 1992, over 300 companies were members of PCMCIA.

4.1.1 Personal Computer Form Factors

Figure 4.1, "Personal Computer Form Factors", shows a range in size of portable computers addressed by the PCMCIA standard. The product segments shown in Figure 4.1 do not include some of the emerging portable products that will use PC Cards such as Personal Digital Assistants and Personal Communicators. In addition, many other non-portable PCs will incorporate memory card technology. Some examples of other products that will have a PCMCIA interface for PC Memory Cards include medical diagnostic equipment for updating analysis algorithms, laser printers for importing new font storage, and electronic diagnostic equipment for reconfiguration of the tester.

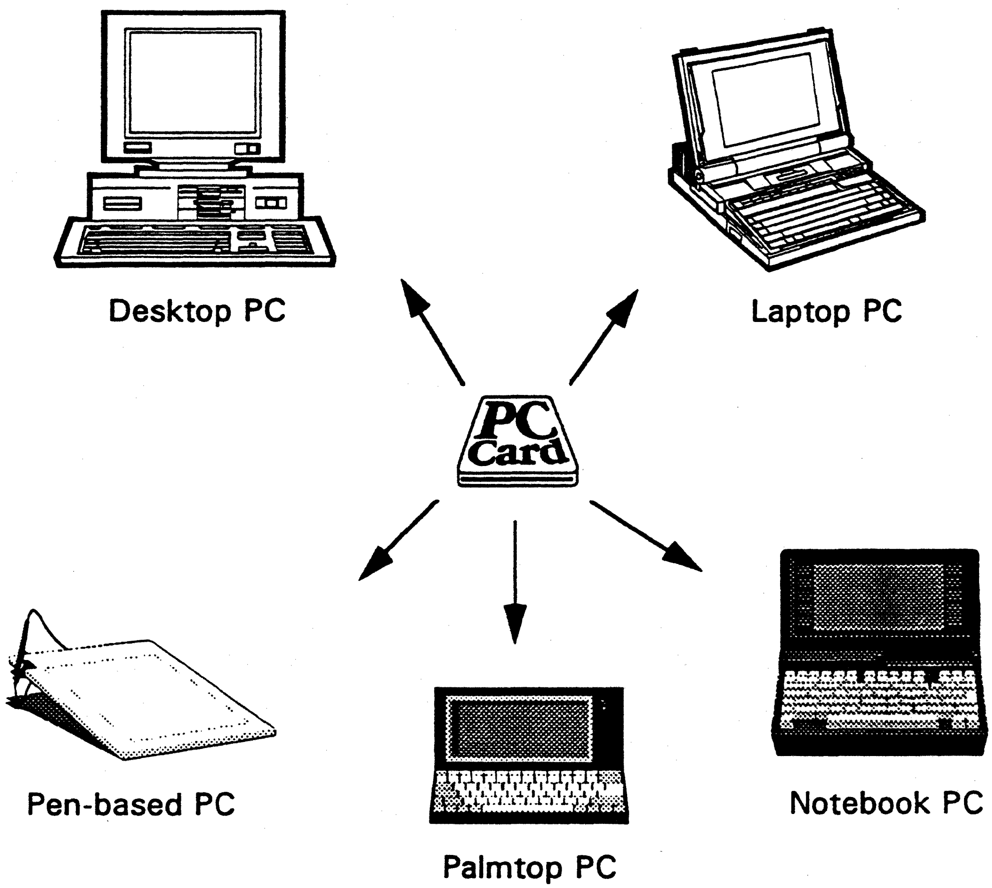

Figure 4.1 Personal Computer Form Factors

4.2 Overview of the Structure of the PCMCIA Elements

The PC Card and its supporting hardware and software can be seen as a series of layers similar to the layers of an IBM Compatible PC. Readers can refer to Appendix B for the layers of an IBM PC. Figure 4.2, "The PCMCIA Hardware and Software Layers", shows all of the major elements necessary to make a Flash memory card operate in an IBM Compatible PC. To generalize the diagram, one should replace DOS with a generic operating system, and eliminate the Memory Technology Drivers (MTDs) and the Flash File System (FFS). With the MTDs and FFS removed, and the operating system generic, the diagram illustrates the general PCMCIA layers for all PC Cards, both Memory and I/O, for use on any host system. The definitions of Memory Technology Drivers (MTDs) and the Flash File System (FFS) will be explored in the later chapters of this manual.

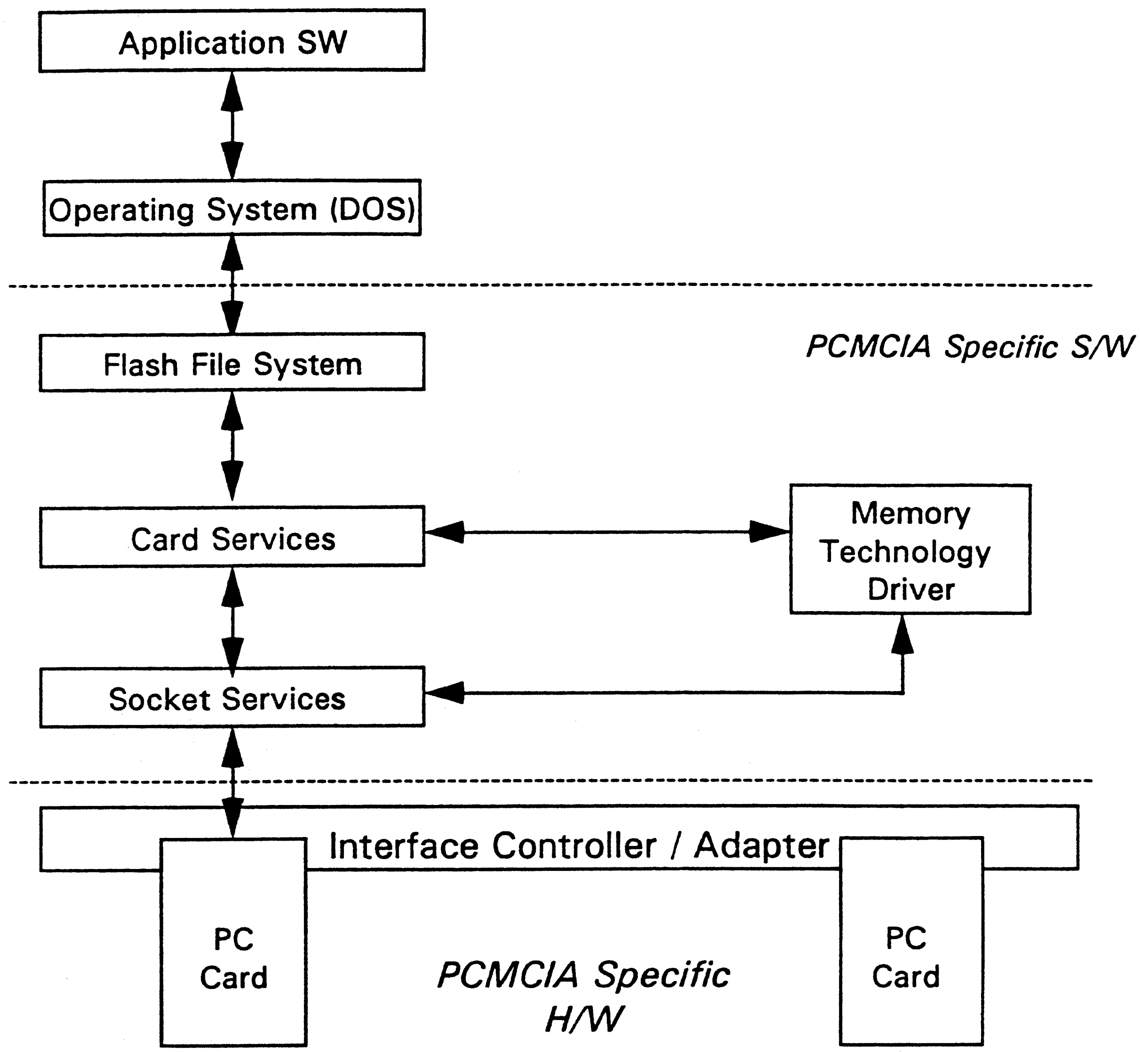

Figure 4.2 The PCMCIA Hardware and Software Layers

Various "blocks" of the above block diagram are provided by different vendors. For example, the "PC Card" could be an AMD Flash memory card. The "Interface Controller/Adapter" could be a TMB200 Card Drive by Databook Inc., or a Solid State Data Drive from Adtron Corporation, or a Databook interface controller chip and a PCMCIA socket. The PCMCIA specific software (i.e. Card Services, Socket Services, and Memory Technology Drivers (MTD)), could all be obtained from a single BIOS vendor or a user could choose to purchase the *Socket Services* software from one vendor and the *Card Services* and the AMD *Memory Technology Driver* software from other BIOS vendors. The *Flash File System* and the application software are available from the software vendors. Flash File System was developed for the Flash Memory devices and it is not needed to access I/O Cards. The Operating System (OS) is usually packaged with the computer. Some of the OS vendors (and their Operating Systems) are: Microsoft (MS DOS), IBM (PC DOS, O/S2), Digital Research (DR-DOS), Apple (Systems 7), and the UNIX Operating System.

To promote greater end user acceptance of the PCMCIA mobile computer concept, hardware manufacturers could provide a one stop shopping for the portable computer user by combining all the elements of the PCMCIA hardware and software layers into one package.

Chapter 5. Understanding PCMCIA Cards and Connectors

5.1 Physical Interface

This section describes the PC card's physical outline, dimensions, and connector systems. There are three different types of PC Cards defined by PCMCIA. Each of the three card types has the same length and width but they differ in thickness. The connector system consists of a female and a male section. The PC Card is the female component and the fixed pin socket on the host PC is the male component. The PC Card, which is *keyed*, and the system mate in such a way that the PC Card cannot be inserted upside down. The keying of the PC Card is illustrated in Figure 5.2, "The PC Card Connector".

5.1.1 Card Dimension

There are three "Types" of PC cards currently defined by PCMCIA: *Type I*, *Type II* and *Type III*. Figure 5.1, "The PC Card Types", shows a three dimensional diagram of each of the three card Types. All three card Types have the same outline, but they differ is thickness. Additionally, PCMCIA has approved an "Extended version" of the PC Cards. The details of the Extended PC Cards will be covered in Volume II.

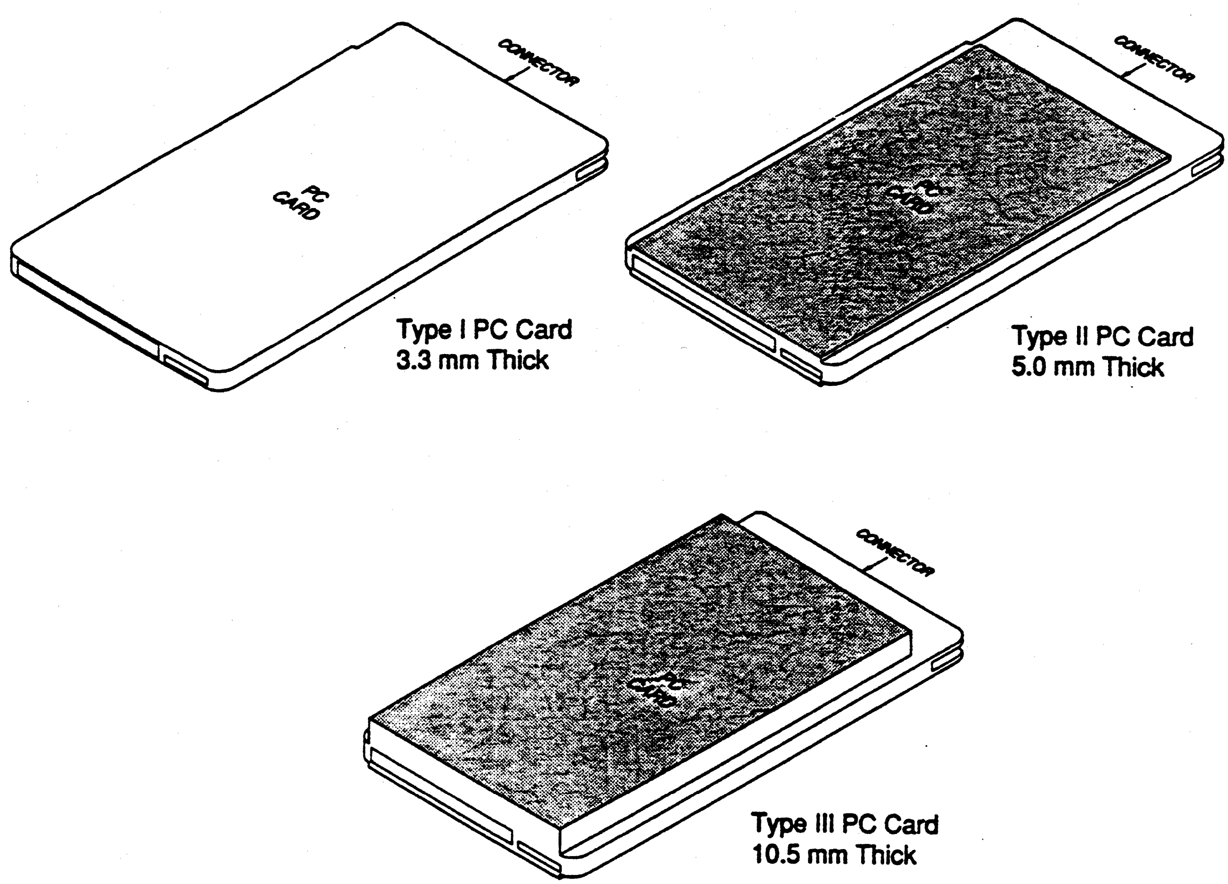

Figure 5.1 The PC Card Types

Table 5.1, "PC Card Dimensions", gives the PC cards dimension for Type I ,Type II and Type III PC Cards.

Table 5.1 PC Cards Dimensions

Card Type	Length inches (mm)	Width inches (mm)	Thickness inches (mm)
Type I	3.37 (85.6)	2.126 (54)	0.13 (3.3)
Type II	3.37 (85.6)	2.126 (54)	0.196 (5)
Type III	3.37 (85.6)	2.126 (54)	0.413 (10.5)

5.2 Connector System

The *PC Card interconnect system* consists of two pieces: the PC Card and the *interface connector* on the host. The interface connector has 68 pins, and the PC card has 68 sockets that correspond to the pins on the connector. Figure 5.2, "PC Card Connector", shows an end view of a PC Card connector and its *pin out*. The left side of the PC Card connector is mechanically keyed differently from the right side, the host interface connector is keyed to accept this configuration and prevents the PC Card from being inserted upside down.

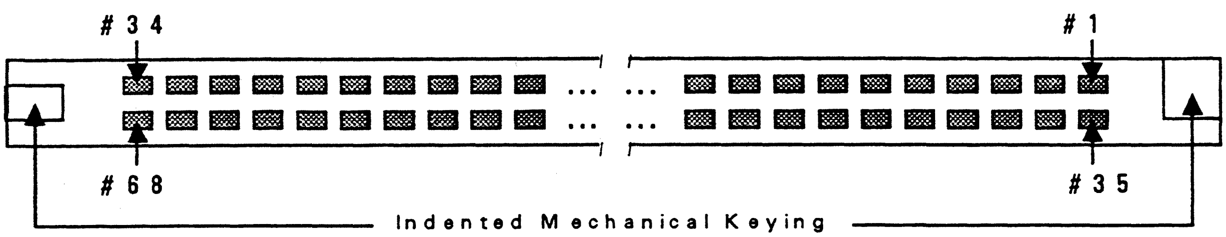

Figure 5.2 PC Card Connector

5.3 PCMCIA Standard: Releases 1.0 and 2.0

PCMCIA Release 1.0 defined all the required hardware and format aspects needed to build standard PC Memory Cards. PCMCIA Release 1.0 defined issues related to handling memory cards used for data storage in various environments. Memory cards used for data storage could use SRAM with battery backup or one of five non-volatile memory technologies: Mask-ROM, EEPROM, EPROM, OTPROM or Flash memory. The inclusion of these five memory types was based on price/performance considerations of the different storage media. In other words, OTPROM Cards might be used in some applications even though they cannot be reprogrammed because they are less expensive than Flash memory Cards.

5.3.1 PCMCIA Standard 2.0: Addition of I/O Card Standard

As PC Memory Cards started to become more common and the use of portable computers increased, manufacturers of modems and Ethernet Cards became interested in PCMCIA. As the PCMCIA standard evolved and gained more widespread adoption, the PCMCIA interface took on characteristics of a standard bus for portable computers. Thus the manufacturers of modems and other I/O expansion cards for desktop PCs to champion the inclusion of I/O PC Cards in the PCMCIA standard.

The desire of I/O expansion card manufacturers to be included in PCMCIA lead to PCMCIA Release 2.0. This Release added some I/O access specific signals to PC Cards in order to support I/O capabilities. In the software area, Release 2.0 added the software specifications for eXecute-In-Place (XIP) and Socket Services 1.01. XIP allows the system to execute application programs directly from the memory card without downloading the actual code to the system RAM. The details of XIP will be discussed in Volume II. Socket Services provides the software protocols to control the system's access to sockets and PC cards. The purpose of Socket Services is introduced in Chapter 6, "PCMCIA Hardware and Firmware". Card Services is introduced in Chapter 7, "Understanding PCMCIA Software Standards". Both Socket Services and Card Services will be discussed in more detail in Volume II of this Training Manual.

5.3.2 PCMCIA Working Groups

PCMCIA is an open standard that can accommodate further enhancements to PC Cards. As PC Cards are used in more and more systems, and as the portable computing industry continues to grow, the likelihood of more features being added to PC Cards increases. The *PCMCIA working groups* are investigating various advanced features such as a "32 bit bus master" and Direct Memory Access (DMA) support to enhance I/O capabilities. A complete listing of the working groups and their activities will be discussed in the "PCMCIA Working Groups" section of Volume II. Since the use of PC Cards continues to grow and PC Card users and manufacturers are asking for enhanced features on the PC Cards, PCMCIA will continue to enhance the Release 2.0 specification to support those future requirements.

5.3.3 PCMCIA Release 2.0 Enhancements and Additions

The PCMCIA Release 2.0 provided for several enhancements and additional features over the Release 1.0 specification. The enhancements and additions are as follows:

- Memory Cards <u>may</u> take advantage of the Release 2.0 RESET/WAIT# signal.
- Addition of I/O hardware and software interface protocol.
- Addition of Dual Voltage Operation for PC Memory Cards that can operate at reduced Vcc voltage.
- Addition of the Geometry Tuple and the Interleaving Tuple. Both tuples are primarily used in support of the Flash memory Cards.
- Addition of the eXecute In Place software specification to allow PCs to execute application programs directly from a PC Card.

5.3.4 Compatibility between Release 1.0 and 2.0

In order to allow the PCMCIA Release 1.0 compliant PC Memory Cards to operate in a PCMCIA Release 2.0 compliant host system, a Release 2.0 compliant host system is required to perform the following:

- Power-up a PC Card in the default power-on state. The default power-on state is: Vcc = 5.0 V, and Release 1.0 "Memory only Interface" protocol.
- Access the Card Information Structure (CIS) of the PC Card to determine if the PC Card is a Release 1.0 or Release 2.0 compliant PC Card.
- If the Card is a Release 2.0 PC Card, determine from the CIS if the card is a Memory Card or an I/O Card.
- If the Card is an I/O Card, activate the Release 2.0 I/O Interface.

The above requirements provide for *backward compatibility* for Release 1.0 compliant PC Memory Cards in a Release 2.0 compliant host system. In addition, a Release 2.0 compliant Memory Card inserted into a Release 1.0 socket must appear to the host system on power-on as a Release 1.0 compliant card in its initial default power-on state.

5.4 Environmental Requirements of PC Cards

There are a cadre of environmental requirements that a PC Card must pass in order to be classified as a PCMCIA compliant card. Interested readers should refer to Section 3 of the PCMCIA Standard which defines the requirements that a PC Card and connector must meet from an environmental standpoint. The list below represents a small number of the environmental requirements that a PC Card must meet. However, this list does give a feel for the environment in which a PC Card and connector may be used.

5.4.1 Connector Mechanical Reliability

The PCMCIA standard guarantees that a PC Card will meet the following minimum mechanical performance:

- Connector Endurance: 10,000 minimum cycles card insertion and ejection in office environment (i.e., a building with year-round air conditioning and humidity control), and 5,000 minimum cycles in a harsh environment (i.e., a building with no air conditioning, and no humidity control, but normal heating and ventilation.)
- Total card insertion force: 8.8 lbs maximum
- Total card removal force: 1.5 lbs minimum
- Vibration: 15G from 10 Hz to 2000 Hz
- Shock: 50G

5.4.2 Connector Electrical Performance

- Current capacity: 0.5 A per pin

5.4.3 Connector Temperature Ranges

- Operating temperature: -20 to +60 degrees C (-4 to 140 degrees F)
- Storage temperature: -40 to +70 degrees C (-40 to 158 degrees F)

5.4.4 General PC Card Environment

- Electrostatic discharge: 1500 V through 1500 ohms resistor.
- Electromagnetic field interference: 1000 Oersted for 10 seconds
- X - ray exposure: 10K Roentgen for 5 minutes

5.4.5 PC Card Temperature Ranges

- Operating temperature: 0 to +55 degrees C (32 to 131 degrees F)
- Storage temperature: -20 to +65 degrees C (-4 to 149 degrees F)

Chapter 6. Understanding PCMCIA Hardware and Firmware

6.1 PCMCIA Hardware

This section gives a system level view of how the PCMCIA hardware interfaces with the host system, and discusses the features of the PCMCIA hardware. The first sub-section discusses how a PC Card interacts with the IBM Compatible PC, and specifically the ISA bus. The second sub-section discusses the features of the PC Card, and describes the *PCMCIA address space* and configuration registers. The third sub-section introduces the PCMCIA signal descriptions. The final sub-section introduces the *PCMCIA Interface Controller*, or *Adapter*, and gives a list of some of the PCMCIA Interface Controllers currently on the market.

6.1.1 System Level Block Diagram

Figure 6.1 shows how the PC Card and the Interface Controller/Adapter fits into the elements of a PCMCIA implementation.

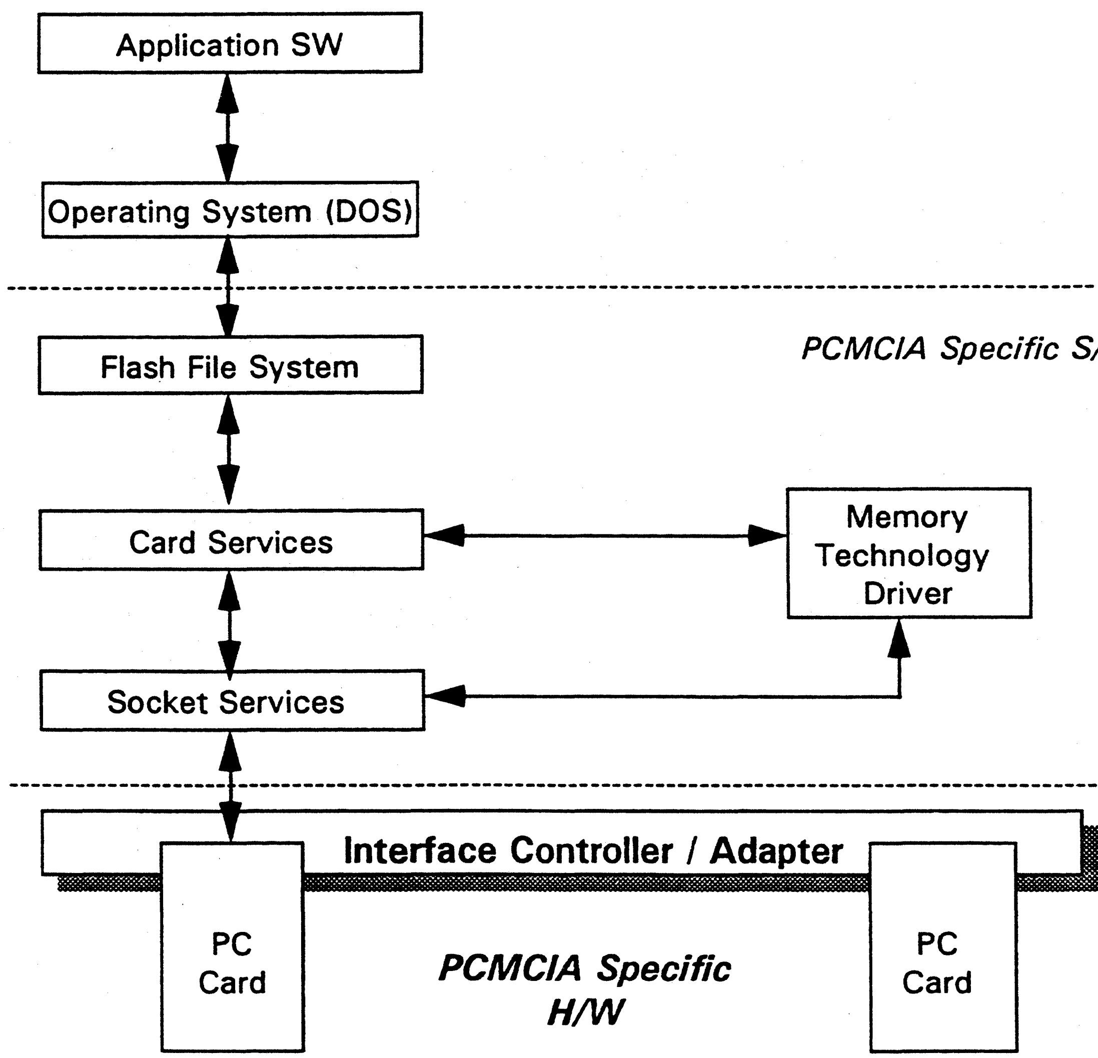

Figure 6.1 The PCMCIA Layers: Focus on PCMCIA Hardware

Figure 6.2 shows how the PCMCIA Interface Controller/Adapter connects with the ISA bus. The diagram shows that the only extra logic required to support the PCMCIA interface design on an IBM Compatible PC is the PCMCIA Interface Controller logic. This piece of hardware is commonly known as the *Adapter*. The Adapter can be implemented with discrete logic or as an integrated chip. The PCMCIA Interface Controller logic sits between the ISA bus of the IBM Compatible PC and the PCMCIA bus and provides the required interface to allow a PC Card to interact with the host system. The Adapter accepts ISA bus compatible signals and generates various control signals for the PC Cards. Any access by the host system to a PC Card is handled through the Interface Controller/Adapter.

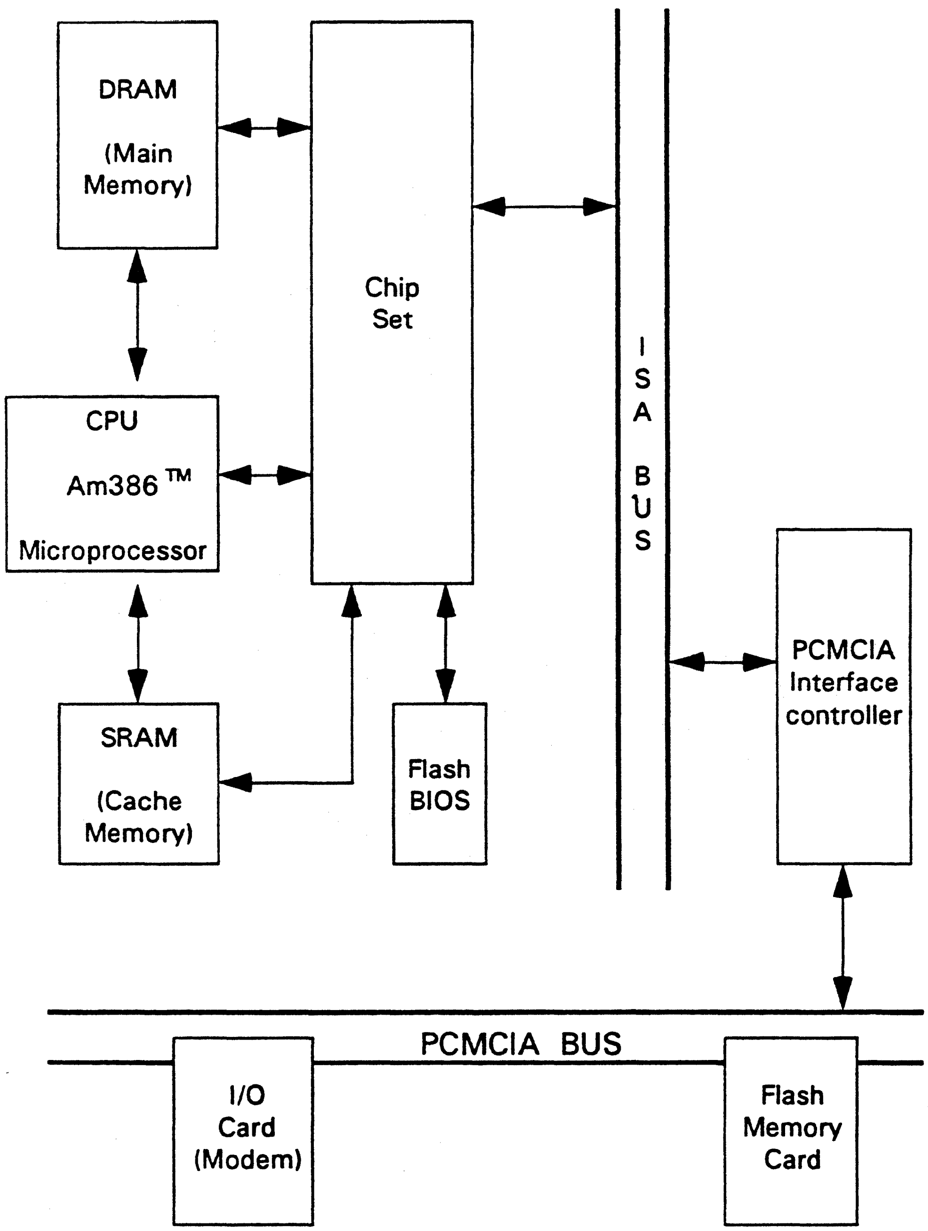

Figure 6.2 PC-AT Block Diagram with PCMCIA Support

6.2 The PC Card

PCMCIA compliant memory cards can contain any of the following six memory devices: SRAM, Mask ROM, OTPROM, EPROM, EEPROM, and Flash memory. PCMCIA defines the following *access speeds* for PC Memory Card operation: 100ns, 150ns, 200ns, and 250ns read cycles for all memory types and 100ns, 150ns, 200ns, and 250ns write cycles for SRAM. Write timing for OTPROM, EEPROM and FLASH based PC Cards is device specific. For *Attribute Memory access*, the PCMCIA defines a universal 300ns read cycle time and 250ns SRAM write cycle time. Please note that Attribute Memory is discussed later in this chapter. A 600ns speed version is defined by PCMCIA standard 2.0 for memory PC Cards that can operate at the Reduced Operating Voltage condition of Vcc equals 3.3 volts.

PCMCIA defines a 68-pin interface and is configured for an 8-bit or 16-bit data transfer. The PCMCIA standard supports both memory cards and I/O cards. Figure 6.3 shows a diagram of the internal layout of two PC Memory Cards. A PCMCIA memory card with its Attribute Memory space inside the Common Memory is known as an aliased memory card.

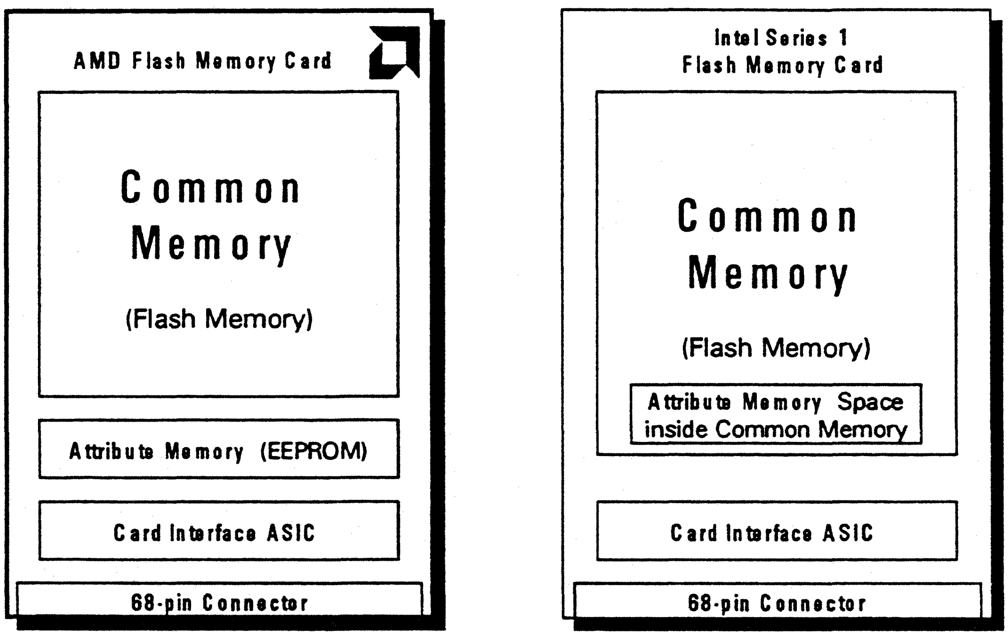

Figure 6.3 The PCMCIA Flash memory Cards

6.3 PCMCIA Address Space

The standard PCMCIA interface (i.e., PC Card and connector when mated) has 26 address lines (A0-A25). Thus, the PCMCIA interface supports 64 Mbytes of address space for both memory and I/O cards. PCMCIA supports three different types of address space:

- Common Memory Space
- Attribute Memory Space
- I/O Space

Common Memory space is the primary address space containing the memory used for data storage. Common Memory space can be a maximum of 64 Mbytes. Common Memory space and the Attribute Memory space are selected per Tables 6.2, "Common Memory Read Function for all types of Memory", Table 6.3, "Common

Memory Write Function for SRAM, EEPROM and Single-Supply Flash memory", and Table 6.4, "Attribute Memory Read Function for all types of Memory".

Attribute Memory space is used to store the PC Card's identification and configuration structure and can be a maximum of 64 Mbytes. Attribute Memory space is logically separate from Common Memory space. Attribute Memory consists of the Card Information Structure (CIS) and optional Card Configuration Registers. The CIS and the Card Configuration Registers are defined later in this chapter. The details about the CIS are contained in Chapter 7, "Understanding PCMCIA Software Standards". Attribute Memory space is logically selected when the REG pin is driven low. Selecting between Common and Attribute Memory space will be discussed further in the later sections.

6.3.1 I/O Address space and the Memory Only Interface

I/O address space is used only in PC I/O Cards and not in PC Memory Cards. Recall, PCMCIA Standard Release 1.0 was developed for PC Memory Cards. PCMCIA Release 2.0 was developed to include functionality for PC I/O Cards. Upon the introduction of PC I/O Cards, it was necessary that the PCMCIA interface be able to distinguish between a PC Memory Card and a PC I/O Card. Thus, the *Memory Only Interface* was defined for PC Memory Cards and it does not contain signals to support PC I/O Cards. In other words, some PCMCIA interfaces are designed for PC Memory Cards only. In order to address the problem of a user putting a PC I/O Card into a Memory Only Interface socket, PCMCIA defined that when a PC Card was inserted into a socket, the Adapter control logic should default to the Memory Only Interface upon Power-up of the PC Card. Thus, the default condition of all PCMCIA sockets is to Power-up in the Memory Only Interface condition to provide backward compatibility with PCMCIA Standard Release 1.0. Even if a system is capable of supporting I/O PC Cards, the socket is always configured to the Memory Only Interface whenever a card is inserted. Once the PC Card is present, the system can read the PC Card's CIS to determine whether there is a memory or I/O card present. If the PC Card has I/O capabilities, then the system can reconfigure the interface appropriately.

6.3.2 The Attribute Memory Space

As mentioned before, the Attribute Memory space is an address space that is selected by driving the REG# pin low. All PC Cards that meet the PCMCIA standard have Attribute Memory space, but the Attribute Memory space could be empty. In other words, the Attribute Memory space is an address space that PCMCIA has defined for storing the card specific information, but one can choose not to store any information in this address space.

The differences between Attribute Memory space and *Attribute Memory* are frequently confusing to someone not familiar with PCMCIA. Attribute Memory space is the logical address space that contains the CIS and the optional Configuration Registers. The Attribute Memory space can be located in the same device(s) that contain Common Memory or it can be stored in a physically separate Attribute Memory (i.e., memory device.) Figure 6.3 shows the difference between a memory card with a separate Attribute Memory and one with the Attribute Memory space as a part of the Common Memory space.

An example of the use of a physically separate Attribute Memory is the current AMD Flash memory Cards, where the CIS is stored in separate EEPROM on the Flash memory Card. An example of the CIS being stored in the same devices as Common Memory is the Intel "Series I" Flash memory Card. In other words, Intel stores the CIS on the Flash memory devices that are part of the Flash memory Card's Common Memory.

Attribute Memory space is accessed by using the Attribute Memory Select (REG#) signal. The "#" after any signal name identifies the signal as an active low signal throughout this manual. Attribute Memory access supports 8-bit wide operation, but the PC Card has a 8-bit or 16-bit wide data bus. In order to support 8-bit wide operations, D0-D7 are the only valid bits during Attribute Memory data transfer. Only the even bytes of data are valid during an Attribute Memory operation. For example, bytes 0, 2, 4, etc. are valid, and bytes 1, 3, 5, etc. should be ignored by the system.

6.3.2.1 The Card Information Structure

The *Card Information Structure* (CIS) contains information about the formatting and data structures of the PC Card. The CIS begins at address 00h of the Attribute Memory space. As mentioned above, more details about the CIS are contained in Chapter 7, "Understanding PCMCIA Software".

6.3.2.2 The Card Configuration Registers

Some PC Cards can be configured by the system. For example, a modem I/O PC Card needs to be configured by the system before it can communicate with another modem. A PC Card that can be configured by the system contains Card Configuration Registers. The Card Configuration Registers are located in the Attribute Memory space and they are mostly used by the I/O PC Cards. Although some PC Memory Cards contain the Card Configuration Registers, those memory cards require special and proprietary software to access the Card Configuration Registers.

The Card Configuration Registers are a set of numbered, standardized, 8-bit wide registers. These registers are addressed at consecutive even byte addresses starting at a "*Base address*" in the Attribute Memory space. A part of the CIS, known as the Card Interface Tuple, tells the host system the Base address of the Card Configuration Registers. Chapter 7, "Understanding PCMCIA Software Standards" gives more information about tuples. The Card Configuration Registers define the electrical interface requirements for the PC Card, the I/O address space for the PC Card, the interrupt request, and power requirements of the PC Card. Since all Card Configuration Registers are used to configure the PC Card, they may be both read and written. A list of Configuration Registers defined by PCMCIA is shown below. Memory cards are not required to but may use all Configuration Registers except the Pin Replacement Register

Configuration Option Register
This register is used to configure the PC Card for a specific type of operation. This register can also be used to issue a soft reset (software controlled reset function) to the PC Card. This register defines the interrupt mode of the card (i.e., pulse interrupt or level interrupt) and is written along with the *Configuration Index* on Power-up of the PC Card. Bit 0 to bit 5 of this Register contain the Configuration Index of the PC Card. When the Configuration Index (D0 to D5 of the Configuration Option Register) is written with 0, the PC Card's I/O capability is disabled and the Memory Only Interface will be used. This register must be implemented in all configurable PC Cards.

Card Configuration and Status Register
This register is used to control the Power-Down function (for Low Power consumption) of the PC Card. This register is also used to configure the PC Card for 8 bit I/O access only.

Pin Replacement Register
This register is implemented to give the status of the Write Protect Switch, the Ready/Busy and the Battery Voltage Detect signals when the PC Card is configured for I/O operation.

Socket and Copy Register
This register is used to provide a unique identification for similar PC Cards in the system.

6.4 PCMCIA Signal Descriptions

The following signals are defined by the PCMCIA standard, Release 1.0. The # sign after any signal name identifies that signal as an active low signal. Please note that the PCMCIA Standard uses a "-" in front of the signal to denote an active low signal.

Signal	Signal Descriptions
A0-A25	Address Bus (Input)
D0-D15	Data Bus (Input/Output)
CE1# and CE2#	Card Enable 1 and Card Enable 2 (Input) The CE1# signal enables the even address bytes and CE2# signal enables the odd address bytes
OE#	Output Enable (Input) This signal has to be active for any read operation
WE#	Write Enable (Input) This signal is activated for any write operation
RDY/BSY#	Ready/Busy (Output) When low, this signal indicates the card is busy with a previous write operation. When high, this signal indicates that the PC Card is ready for access
CD1# and CD2#	Card Detect 1 and Card Select 2 (Output) These two signals provide for the proper detection of PC Memory Card insertion
WP	Write Protect (Output) This signal reflects the status of the Write Protect switch
REG#	Attribute Memory Select (Input) This signal is used to select the Attribute Memory access and it is always active for an I/O access
BVD1# and BVD2#	Battery Voltage Detect (Output) The signals BVD1# and BVD2# are generated by the PC Memory Card as an indication of the condition of the battery
VCC & GND	Card Voltage and Ground The VCC must be within the operating voltage range (e.g. 5 Volts) when a card is read initially.
VPP1 and VPP2	Program and Peripheral Voltages The VPP1 and VPP2 signals supply programming voltages for programmable memory card or additional supply voltages for peripheral cards.

Note: The two signals WAIT# and RESET listed below are defined by PCMCIA Release 2.0. Memory cards and systems compatible with Release 2.0 are required to support these two signals.

Signal	Signal Descriptions
RESET	Card Reset (Input) This signal resets the PC Card into a known state that is the same as the power-up default condition
WAIT#	Extend Bus Cycle (Output) This signal is used to generate any wait states to a memory or an I/O cycle

The following signals are defined by the PCMCIA standard, Release 2.0 to support I/O cards.

I/O Signal	Signal Descriptions
IORD#	I/O Read (Input) This signal is activated in conjunction with the REG# signal and either of the CE1# signal or the CE2# signal to read data from the PC Card's I/O space
IOWR#	I/O Write (Input) This signal is activated in conjunction with the REG# signal and either of the CE1# signal or the CE2# signal to write data to the PC Card's I/O space
INPACK#	Input Acknowledge (Input) This signal is used by the host system to control the enable of any input data buffer between the PC Card and CPU
IOS16#	16 Bit I/O (Output) This signal indicates a 16 bit I/O access and it replaces the WP signal in a Memory Only Interface
IREQ#	Interrupt Request (Output) This signal is used by a PC I/O Card to generate an interrupt request to the host system. This signal replaces the RDY/BSY# signal in a Memory Only Interface
SPKR#	Audio Digital Wave form (Output) This signal replaces the BVD2# signal in a Memory Only Interface
STSCHG#	Status Changed (Output) This signal is used to indicate to the system any changes in the Ready/Busy (RDY/BSY#) signal, the Write Protect (WP) signal, or the Battery Voltage Conditions (BVD) signal of the PC Card, while the I/O interface is configured

6.4.1 PC Card Logic Levels

Signals on the PCMCIA Interface, whether operating at the standard operating supply level of 5.0 volts or the reduced operating supple voltage of 3.3 volts, are considered asserted within the range of VOH and negated within the range of VOL. However, the VOH and VOL ranges are determined by the supply voltage as show in Table 6.1 below. VOH, VIH, VOL, and VIL for 3.3V operation are under review in JEDEC.

Table 6.1 PC Card Logic Levels

DC Levels		System Host		
Parameter	Conditions	Min.	Max.	PC Card
VIH	Vcc = 5 V +/- 5%	2.4 V	Vcc + 0.25V	TTL or CMOS
VIL	Vcc = 5 V +/- 5%	0.0 V	0.8 V	TTL or CMOS
VOH	Vcc = 5 V +/- 5%	2.8V (0.9 *Vcc) [1]	Vcc	TTL or CMOS
VOL	Vcc = 5 V +/- 5%	0.0 V	0.5V (0.1*Vcc) [1]	TTL or CMOS

Note 1: for CMOS Loads (High input resistance)

6.4.2 PCMCIA Common Memory Read/Write Function

The Read function of the PC Memory Cards shares a common signal-state sequencing. To access the Common Memory, the signal REG# must be keep inactive, and the signal OE# must be active during the Read Cycle. Signal CE1# and CE2# control the activation of Word-wide or Byte-wide access of the PC Memory Card and address line A0 controls the byte ordering on the data-bus lines D0 to D15. Table 6.2, "Common Memory Read Function for all types of Memory" and Table 6.3, "Common Memory Write Function for SRAM, EEPROM, and Single-Supply Flash memory" show the signal state and the valid data conditions for the Common Memory Read Function and Write Function, respectively. PCMCIA compliant systems that are based on 8-bits microprocessor should support "Byte access" and "Odd-byte only access" in tables 6.2, 6.3, and 6.4..

Table 6.2 Common Memory Read Function for all types of Memory

Function Mode	REG#	CE2#	CE1#	A0	OE#	WE#	D15-D8	D7-D0
Standby Mode (High-Z)	X	H	H	X	X	X	High-Z	High-Z
Byte access (8 bits)	H	H	L	L	L	H	High-Z	Even-Byte
	H	H	L	H	L	H	High-Z	Odd-Byte
Odd-byte only access	H	L	H	X	L	H	Odd-Byte	High-Z
Word access (16 bits)	H	L	L	X	L	H	Odd-Byte	Even-Byte

Table 6.3 Common Memory Write Function for SRAM, EEPROM and Single-Supply Flash memory

Function Mode	REG#	CE2#	CE1#	A0	OE#	WE#	D15-D8	D7-D0
Standby Mode (High-Z)	X	H	H	X	X	X	High-Z	High-Z
Byte access (8 bits)	H	H	L	L	L	L	High-Z	Even-Byte
	H	H	L	H	L	L	High-Z	Odd-Byte
Odd-byte only access	H	L	H	X	L	L	Odd-Byte	High-Z
Word access (16 bits)	H	L	L	X	L	L	Odd-Byte	Even-Byte

Notes: X = Don't Care High Z = High Impedance
L = VIL H = VIH

6.4.3 PCMCIA Attribute Memory Read Function

To access the Attribute Memory, the signals REG# and OE# must be active during the Read Cycle. As in the Common Memory Read Function, Signal CE1# and CE2# control the Even-byte and the Odd-byte address, but only the Even-byte data is valid during the Attribute Memory Read Function. Table 6.4, "Attribute Memory Read Function for all types of Memory" shows the signal state and the valid data conditions for the Attribute Memory Read Function.

Table 6.4 Attribute Memory Read Function for all types of Memory

Function Mode	REG#	CE2#	CE1#	A0	OE#	WE#	D15-D8	D7-D0
Standby Mode (High-Z)	X	H	H	X	X	X	High-Z	High-Z
Byte access (8 bits)	L	H	L	L	L	H	High-Z	Even-Byte
	L	H	L	H	L	H	High-Z	Not Valid
Odd-byte only access	L	L	H	X	L	H	Not Valid	High-Z
Word access (16 bits)	L	L	L	X	L	H	Not Valid	Even-Byte

6.5 PCMCIA Interface Controller/Adapter Vendors

The PCMCIA Interface Controller logic for the host system is currently provided by a number of suppliers. The following list gives some of the available PCMCIA Interface Controllers and their manufacturers.

(1) DB86081 (for Palmtop Computers) from Databook Inc. and Fujitsu
(2) DB86082 (for Notebook Computers) from Databook Inc. and Fujitsu
(3) MB86301 from Fujitsu
(4) i82365SL from Intel
(5) J6942 from ASCII
(6) F8680 PC single chip PC with PCMCIA 1.0 support from Chips and Technologies
(7) VG-230 PC-XT on a chip with PCMCIA 2.0 support from Vadem
(8) VL82C107-LT SCAMP PC peripherals with PCMCIA 1.0 support from VLSI Technology
(9) CL-PD6710 and CL-PD6720 from Cirrus Logic

6.6 PCMCIA Firmware: Socket Services

Socket Services is the lowest layer of the multi-layer software architecture that manages PC Card resources. Socket Services is a hardware specific software interface used to manage PC Card sockets in a host computer. Figure 6.5, "The PCMCIA Layers: Focus on Socket Services", shows how Socket Services interacts with the other PCMCIA elements. Socket Services sits on top of the PCMCIA Hardware and is expected to be supplied as *Firmware*, although it may be offered as a *loadable device driver*. If offered as firmware, Socket Services could be stored in a Flash memory device just like the system BIOS. In fact, it is likely that Socket Services will be stored in the same Flash device as a BIOS extension.

Socket Services is the only software layer which is hardware dependent. Thus, Socket Services is a unique implementation dependent upon the Interface Control logic and must be updated with any change in the host interface hardware. Socket Services makes all other software components, including Card Services, MTDs, and the Flash File System independent of the host system hardware.

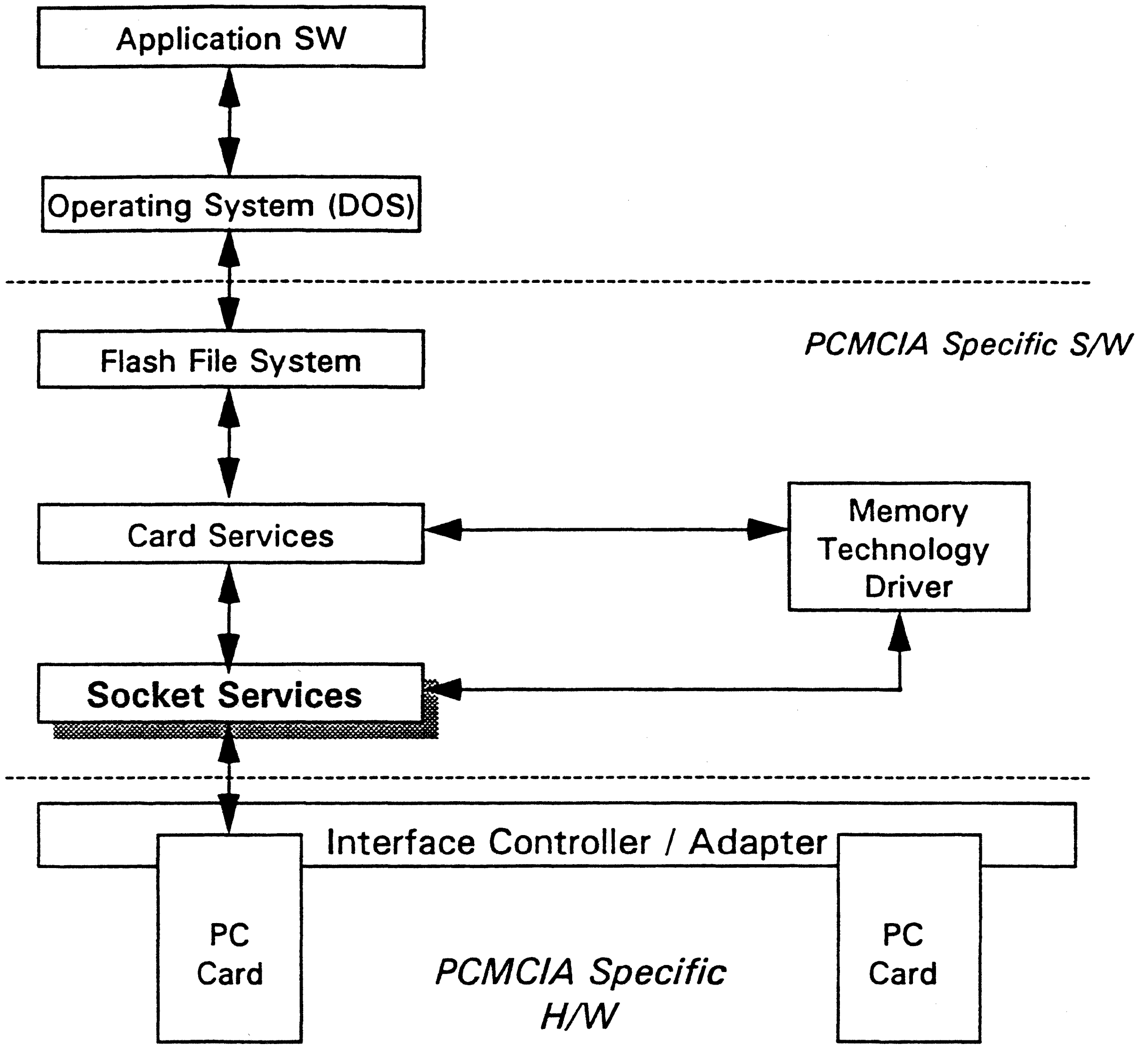

Figure 6.4 The PCMCIA Layers: Focus on Socket Services

6.6.1 The Functions of Socket Services

Socket Services provides a *software function set* that controls the PCMCIA Adapter or interface logic. The *PCMCIA Adapter* is a piece of hardware which interfaces between the host computer's address and data bus (ISA Bus) and the PC Card(s). An Interface Controller/Adapter may have one or more sockets. Socket Services describes the characteristics of each socket and allows the system to determine the current setting of each socket.

Socket Services reports the following information to the system: the number of sockets, the number of windows (not to be confused with Microsoft Windows), and the Error Detection Code (EDC) generators provided by each installed adapter. The *number of sockets* is the number of physical locations that a PC Card can be inserted. A *window* is used to map a portion of the PC Card's 64 Mbytes of Common Memory space, or Attribute Memory space, or I/O space into the host system memory. The details of Socket Services are discussed in Volume II of this manual in the "PCMCIA Firmware: Socket Services" chapter.

Socket Services also reports on current card status and allows data to be read from, or written to PC Cards which are not mapped into system memory. However, for better performance, it is often beneficial to map the PC Card's memory space or I/O space into the host system memory. EXecute In Place (XIP) applications require that PC Card Memory be mapped into system memory space. PCMCIA Interface Controllers/Adapters may or may not provide this capability. If available, mapping is performed through Socket Services, which manipulates the memory windows.

6.6.2 Implementation of Socket Services

As mentioned above, Socket Services will be provided as a ROM BIOS extension or a loadable device driver. However, a ROM extension will be the preferred method, for it will allow the system to boot from a PC Memory Card. Socket Services is a relatively small piece of code and should be easy to integrate into the Flash BIOS. Socket Services uses the PCMCIA Interface Controller's registers (not to be confused with the PC Card's Configuration Registers) for the identification and configuration of PC Cards by reading from, and writing to, these registers.

Socket Services is handled through software interrupt 1Ah. A client/application simply sets the host microprocessor registers for the function desired and then executes the INT 1Ah instruction to access the Socket Services functions.

Currently, Socket Services is defined for *Real mode* (i.e., the DOS environment), and in future releases, Socket Services is expected to support the *Protected mode*.

Finally, there could be more than one implementation of Socket Services for a system. For example, a system with two PCMCIA Adapters may choose to have one Adapter configured for I/O PC Cards only and the other Adapter configured for Memory PC Cards only. In this scenario, there could one Socket Services for the I/O interface only Adapter and another for the Memory Only Interface Adapter.

PART III - PCMCIA Software and Flash memory Cards Access

Part III of this Training Manual introduces the readers to the software terminology of PCMCIA and describes the role of system-level software in the PCMCIA world. This part also describes how a PC Card user would incorporate a PCMCIA card in his or her system if it didn't include the system-level software support from Card Services.

Chapter 7, "Understanding PCMCIA Software", discusses the issue of interoperability of PC Cards across different host systems, and serves as an introduction to the concept of Metaformat and the Card Information Structure. This chapter also discusses the next highest layers of PCMCIA software: Card Services and Memory Technology Drivers, and the Flash File System.

Chapter 8, "Accessing Flash memory PC Cards", describes how a PC Card user would access the data stored on a Flash memory PC Card before the installation of Card Services.

Chapter 7 Understanding PCMCIA Software Standards

7.1 The Need for PC Card Interoperability

One of the most important features that the writers of the PCMCIA specification wanted to ensure was the ability for PC Cards to be used in a variety of host systems. In addition, if a PC Card did not work in a particular host system, it was important that the insertion of the PC Card did not damage the host system or the PC Card. The term "*graceful rejection*" is used by PCMCIA to describe this requirement. Compliance with the mechanical and electrical interface defined by the PCMCIA Specification Release 1.0 or Release 2.0 will allow a host system to "gracefully reject" a PC Card that will not function in the system. Of course, this does not cover the situation in which a PC Card user forces a 10.5 millimeter Type III card into a 3.3 millimeter Type I socket opening.

The Metaformat section in both Releases of the PCMCIA Specification provides the host system a means to find out the type of PC Card inserted into its PCMCIA socket by processing the Card Information Structure. However, the PCMCIA Specifications do not provide the means to implement this concept of interoperability. Card Services and Socket Services are the two PCMCIA software specifications that can provide system specific interoperability for PC Cards. Interoperability, *interchangeability*, and *"plug-and-play"* all refer to the ability to use a PC Card between different types of host systems. A good example of product interoperability is the music cassette tape. No matter which tape player was used to record the music onto the cassette, the cassette can be played in any cassette tape player.

There are at least two different levels of interoperability. First, *absolute interoperability* means that any PCMCIA compliant system could access any PC Card. Absolute interoperability would require that all systems with a PCMCIA socket, from an Automatic Teller Machine to the HP95LX palmtop PC, have the capability to interpret all other systems' software programs and data files. Although some users might find it desirable to have this absolute interoperability, the system overhead required to implement this data/file translation for all operating systems and microprocessors would be tremendous. Obviously, absolute interoperability has advantages, but it also has pitfalls. Second, *system specific interoperability* means that a PC Card could be used on all compatible systems. For example, a PC Card that is usable on all IBM Compatible PCs has system specific interoperability. Clearly, system specific interoperability is highly desirable and is possible to be implemented without the enormous overhead associated with absolute interoperability. System specific interoperability could be realized by implementing the PCMCIA Socket Services 2.0 and Card Services 2.0 specifications. Socket Services was discussed in the last chapter, and Card Services is discussed later in this chapter.

7.1.1 The Benefits of Card Services

In order to provide software drivers to access a PC Card without Socket Services and Card Services in an IBM PC Compatible system, software vendors would have to know many details about the system. Some questions that software vendors would need answered are:

- What kind of PCMCIA socket controller is the system using?
- Has the system implemented Socket Services 1.0 to control the socket or does the system use some other type of socket device driver?
- Does the system's manufacturer have plans to comply with Card Services and Socket Services 2.0?
- Does the system include a Flash File System as an alternate method to access Flash memory Cards?
- Which manufacturers' PC Cards will the system support?
- Does the system support both I/O cards and memory cards, or only I/O cards?

The answers to these questions will determine the scope of the software drivers and could limit the use of the drivers in other systems. Without Card Services, software vendors have to develop specific PC Card programming drivers for each system. This would be like having to develop a word processing program for each PCMCIA system, even if the differences across systems was as simple as having different Interface Controllers. Obviously, the cost associated with this type of development would be much greater than a single word processor for all systems using DOS. The same is true of device drivers that would have to be developed for each unique system.

The next several sections in this chapter give a brief explanation of how Metaformat, the Card Information Structure, Card Services, and Memory Technology Drivers combine to provide the end users with system specific interoperability.

7.2 Metaformat

The PCMCIA *Metaformat* is a universal software structure that all PCMCIA systems should be able to interpret. It describes the contents, layout, and interpretation of the PC Card's Card Information Structure (CIS). Recall from Chapter 6 that the CIS is contained in Attribute Memory. The Metaformat simply tells the system software how the CIS is organized so that the system can access it. The CIS is made up of sets of information about the PC Card known as *tuples*.

Attribute Memory space is a separate memory address space for storing the Card Information Structure, Configuration Registers, and a Reserved Area for future use. In addition, the timing parameters of the Attribute Memory device(s) may be different from that of Common Memory. The reader should refer to Chapter 6 for more details about the PC Card's addressing scheme.

7.2.1 The Need for Metaformat and System Specific Interoperability

Most of the mobile computing platforms today, from the AT&T Safari to the Zenith Laptop, are IBM Compatible PCs. By 1992, the installed base of IBM PCs and PC Compatibles numbered over 60 million systems in US. and 135 millions world wide. The floppy disk from any one of those portable IBM Compatible PCs is compatible with any other IBM Compatible PC whether it be a portable or a desktop system. The PC users of today are accustomed to the concept of interoperability of floppy disks between Personal Computers with compatible Operating Systems.

However, some computer users continue to attempt to use a DOS-formatted floppy disk in an Apple Macintosh, sometimes with disastrous results when the user initializes his or her DOS diskette in the Macintosh and loses all the data on the diskette.

The emergence of the potential wide scope of applications for PC Cards means the ability to read non-DOS cards on DOS-based systems and vice versa will be of significant value to mobile and desk top users. Metaformat provides a universal way for PCMCIA compliant systems to read the PC Card's Card Information Structure (CIS). In other words, the CIS will allow a PC Card user's system to at least understand the "language" of the PC Card data structure, even if the system cannot use the card.

For example, if a user inserted a card containing data with the DOS File structure into an Apple Macintosh computer, processing the CIS will let the Apple Operating System knows that this PC Card has data files from an incompatible system. Although most Macintosh computers can not read the files in this PC Card, the user will be warned that the card contains a valid DOS File data structure and all data would be lost if the card is initialized. There are a few Macintosh computers in use today that contain a special disk drive that can understand files created by the IBM PC, so it is possible to have interoperability across different operating systems and hardware architecture.

7.2.2 Metaformat Compliance Requirements

The Metaformat is a hierarchy of five layers, Layer 0 to Layer 4. The layered structure of the Metaformat pertains to the types of information conveyed to the system. The "PCMCIA Hardware and Software Layers" which are related to the PC hardware and software architecture should not be confused with the Metaformat layers. The Metaformat is one level of detail below the "PCMCIA Hardware and Software Layers". There is a separate PCMCIA ATA specification and specific tuples related to I/O Cards that address the need of the I/O Cards.

Layer 0: The hardware layer

Layer 0 defines the 68-pin electrical interface. All manufacturers of PCMCIA compliant systems or PC Cards must comply with this basic physical layer. PC Card manufacturers can add successively higher layers of Metaformat to add functionality to their PC Cards and to address particular market needs.

Layer 1 to layer 4: The software layers

The information contained in a PC Card's CIS is organized according to these software layers. To comply with Layer 1 to Layer 4, a PC Card is required to contain a Card Information Structure or to have a *pseudo-floppy* data organization as viewed by the DOS operating system. A PC Memory Card with a pseudo-floppy data organization mimics the data organization of a IBM PC floppy disk and can be accessed by any PC using DOS as its operating system. Details of the pseudo-floppy data organization is covered in Section 5.5.3 of the PCMCIA Standard.

Layer 1: The Basic Compatibility Layer

Layer 1 contains the minimal information needed to access the PC Card such as the access speed of the PC Card, the quantity and the type of device(s) in the card. The DEVICE INFORMATION TUPLE (CISTPL_DEVICE) for the Common Memory is required to be the first block of information stored in this layer. It is a Layer 1 tuple that provides the minimal card access information.

Layer 2: The Data Recording Format layer

Layer 2 specifies how the data is organized in the PC Card such as which method is used for the data blocking and the error checking of the data. This layer is analogous to the physical format of a floppy disk but allows for a mixed format card. An example of a mixed format card is a PC Memory Card with SRAM and Flash memory devices inside the card. Data need not be organized in the DOS format to comply with this layer.

Layer 3: The Data Organization Layer

This layer defines how the data is organized on the card. Some of the possible data organizations are:

- DOS file system (or other operating systems)
- Flash File System
- Execute-In-Place (XIP)
- Application specific organization

Layer 4: The System Specific Layer

Layer 4 allows users to define their own data organization for custom applications.

A hypothetical example of Metaformat compliance is a Personal Digital Assistant (PDA) manufacturer that might want to implement the fastest and lowest cost solution possible by complying only to the layer 0 in the physical hardware area. In the system software area, the PDA might only have support for the AMD 5.0 volt-only Flash memory Card and a fax/modem card. Additional support for other types of Memory or I/O cards could be added later by the PDA user as needed. The layered structure of Metaformat allows for both cost containment through product simplicity and product differentiation through the addition of higher layers of Metaformat.

More details about the Metaformat hierarchy is available in the next volume of this manual in the "PCMCIA Software: Metaformat" section.

7.2.3 The PCMCIA Metaformat numeration convention

The PCMCIA Metaformat section is a software documentation that is intended to be used by software programmers. As in most documentation intended for the software audience, the PCMCIA Metaformat section uses a hexadecimal ($base_{16}$) numerical system. Throughout the PCMCIA Metaformat section and in this Training Manual, a number or a letter (A - F) followed by a "h" indicates a hexadecimal number. The only point of confusion that often arise from the use of the hexadecimal system is that the hexadecimal system starts counting at a 0h, not a 1.

For example, in the tuple example in a later section of this manual, 3h is equal to 4 counts and 7h is equal to 8 counts. An other example is that data-bit 7 (D7) is the 8^{th} data-bit and data-bit 0 (D0) is the 1^{st} data-bit of a byte. A table of decimal ($base_{10}$) versus hexadecimal ($base_{16}$) counting is showed here to help the readers to see the different between those two numerical systems.

Table 7.1 Hexadecimal ($Base_{16}$) versus Decimal ($Base_{10}$) Counting

$Base_{10}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Count:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$Base_{16}$	0h	1h	2h	3h	4h	5h	6h	7h	8h	Ah	Bh	Ch	Dh	- Dh	Eh	Fh
Count:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

7.3 The Card Information Structure

The *Card Information Structure* (CIS) is a variable-length linked list of data blocks containing information about the formatting and organization of the data located on the card. Because of the Metaformat structure described above, the Card Information Structure can be read by any Operating System. Processing the CIS provides the system with the necessary information to access the PC card. Some of the card information stored in the Card Information Structures are:

- The date and time the card was initialized
- The card battery-replacement date and time (for SRAM cards only)
- Vendor specification information
- JEDEC programming information for Common Memory and Attribute Memory

Please refer to Section 5 of the PCMCIA Specification for further reference about the CIS information.

7.3.1 The Location of the Card Information Structure

The Card Information Structure always begins at location 00h of the Attribute Memory Space. Recall from Chapter 6, the Attribute Memory data can be located in Common Memory or in a physically separate device on the card. Remember, however, that Attribute Memory is both physically and logically separated from the Common Memory space. Figure 7.1 "Attribute Memory Space as a Part of the Attribute Memory", and Figure 7.2, "Attribute Memory Space as a part of the Common Memory Space", illustrate the distinction between the two locations for Attribute Memory data.

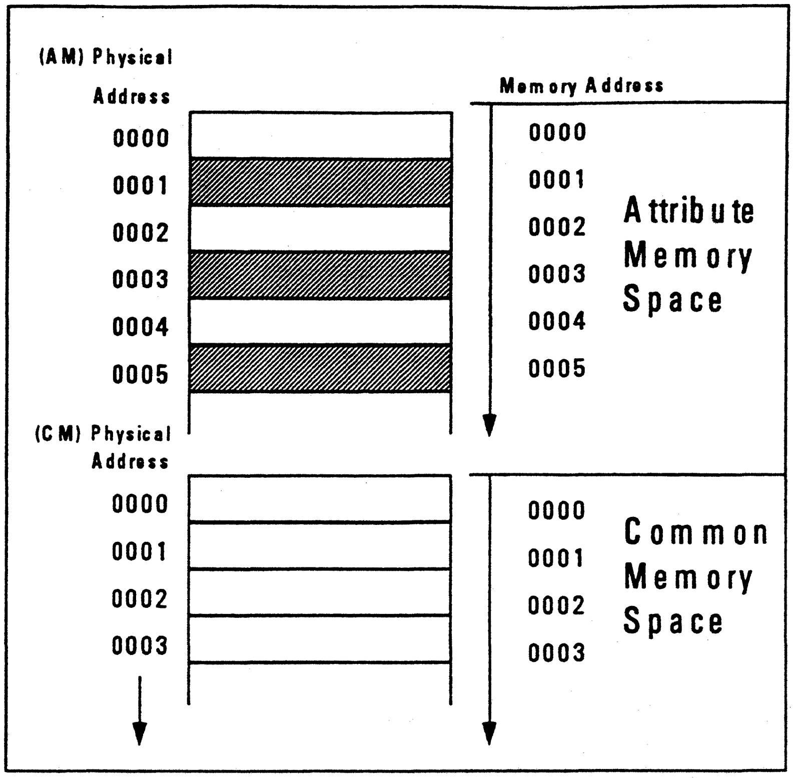

Figure 7.1 Attribute Memory Space as a part of the Attribute Memory

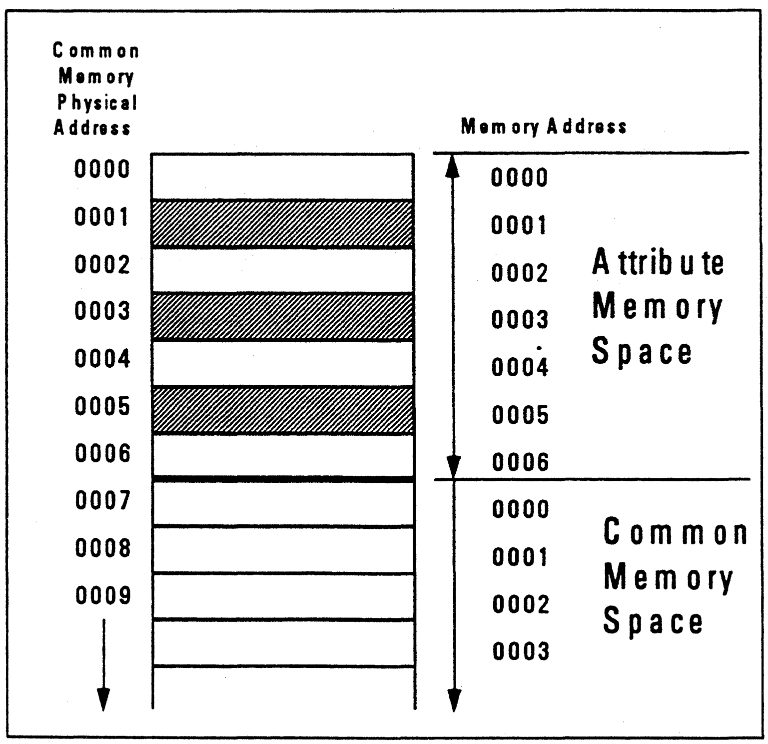

Figure 7.2 Attribute Memory Space as a part of the Common Memory Space

7.3.2 Accessing the Card Information Structure

Regardless if the Card Information Structure is located in a separate memory device or in the Common Memory, the CIS is always accessed by setting the REG# pin (pin 61) low. The Card Information Structure is stored only in even bytes. When Attribute Memory is accessed, only data signals D0 - D7 are valid and signals D8 - D15 will be ignored by the system. Odd byte data stored in the Attribute Memory Space is not defined. Due to the way that the card electrical interface is defined, and to allow 8-bit only systems to access the Attribute Memory, the CIS uses only the even bytes. A detail explanation of the electrical interface for data accessing is beyond the scope of this chapter, but the interested reader should refer to the "PCMCIA Software: Metaformat" chapter in Volume II for more information.

7.3.3 Tuples

A *tuple* is a block of data that appears in the Card Information Structure. All tuples share a common linked list format. In order for the system to know that it has processed a valid tuple, the first tuple must be the Device Information Tuple (CISTPL_DEVICE = 01h), the Null Tuple (CISTPL_NULL = 00h), or the End-of-List tuple (CISTPL_END = FFh). The PCMCIA standard defines which tuples may have fixed length and which tuples may have a variable length.

7.3.3.1 Tuple Format

Tuples are stored in even bytes only in the Attribute Memory space. If a tuple is accessed from the Common Memory space, it must be stored in both even and odd bytes following a long-link target. A long-link target is like a pointer that tells the system software that the tuples are located in address XXXXh of the Common Memory.

- The first byte of a tuple (TPL_CODE) tells the system what type of information this tuple provides.
- The next byte links to the next tuple by telling the system how far away the next tuple is located from the link byte's current address. This *link byte* (TPL_LINK) is a hexadecimal number which starts the count at a 0, not a 1.
- The last byte of a tuple could be FFh to mark the end of data in specific tuples, or it could be the last data byte of that tuple. FFh is required on some tuples such as the Device Information Tuple (01h) and the Product Information Tuple (15h) to mark the end of the tuple.
- The last tuple is always FFh. The End-of-List tuple (CISTPL_END = FFh) marks the end of a tuple chain which marks this tuple as the last tuple in this tuple chain.

This could be confusing to the readers because FFh could represent two meanings. FFh could mark the last byte of data in specific tuples, or it could signify the last tuple of a series of tuples. Additionally, FFh is not required for all tuples. The confusion is eliminated by looking at the link-byte. The link-byte points to the location of the next tuple and at the same time provides the tuple length information. For example, for a tuple stored in the Attribute Memory with a link-byte of 03h, the next tuple is located at four even addresses away from the location of this link-byte (03h in hexadecimal = 4 counts, i.e. 0h-1h-2h-3h). Since the next tuple is four addresses away, one can conclude that there are three more bytes of data in this tuple. If the third byte in this data field is a FFh, then it is the end of data byte, not the last tuple of a series of tuples.

All multi-byte data within the tuple are stored in *little-endian* order with the least-significant byte stored in the first byte of a given tuple data block.

7.3.3.2 Tuple Processing

Like Metaformat compliance, system software can be selective in which tuples to recognize and to act upon. The linked list structure of the tuple format allows the system software to sample and skip over the entire Card Information Structure. If the system software does not recognize a tuple, then that tuple is ignored by the system. The system software reads the next byte, the link byte, to compute where the next tuple is located and jumps to that address. This sample-and-jump process is repeated until the system comes across a tuple that it recognizes.

7.3.3.3 Examples of Processing a Tuple

The two simplest tuples to process are:

1. CIS that only contains the Null tuple (00h)
2. CIS that contains only the End-of-List tuple (FFh)

In either case, the system only processes a single byte of data.

The following example illustrates the third simplest of all tuples to process, the processing of the Device Information tuple. In the example, the PC Card is a Flash memory Card with a 200ns access speed. Figure 7.3, "Tuple Example", shows the actual Attribute Memory tuple codes used. The example is as follows:

- Tuple data are stored in even bytes only in the Attribute Memory.
- The 1st tuple processed is the Device Information tuple for the Common Memory.
- The 2nd byte serves as a link to the next tuple (i.e., in this case the next tuple starts 4 valid bytes ahead of the current location.)
- The 3rd and 4th byte contain data for the DEVICE INFORMATION TUPLE. After decoding, the system understands that the common memory contains Flash memory with a 200ns card access speed and Card Size of 1 Megabyte.
- the 5th byte is FFh to show that it is the last data byte in this tuple.
- The last tuple is always the End-of-List tuple FFh, which tells the system this is the last tuple in the CIS.

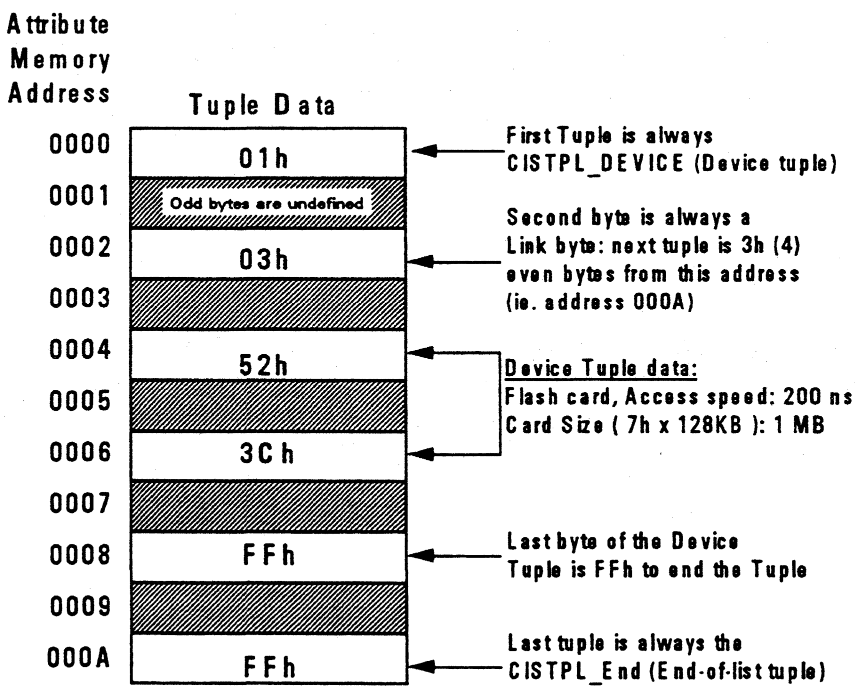

Figure 7.3 Tuple Example

7.4 Card Services

Card Services is a software interface specification that allows PCMCIA-aware device drivers, configuration utilities and application programs to share PC cards, sockets and system resources. For example, Microsoft's Advanced Power Management software for mobile computers contains PCMCIA-aware device drivers. One of the Advanced Power Management's function is to power-off the PCMCIA socket after a pre-determined period of socket idle time. In an IBM compatible PC, Card Services is installed as a device driver at boot time. The Card Services specification is not strictly defined for X86 microprocessors and the DOS operating system. Other computers and operating systems can take advantage of the Card Services functionality. However, Card Services is implemented uniquely for a particular operating system.

Figure 7.4 shows the configuration of Card Services in an IBM Compatible PC using a Flash File System for accessing a Flash Memory card.

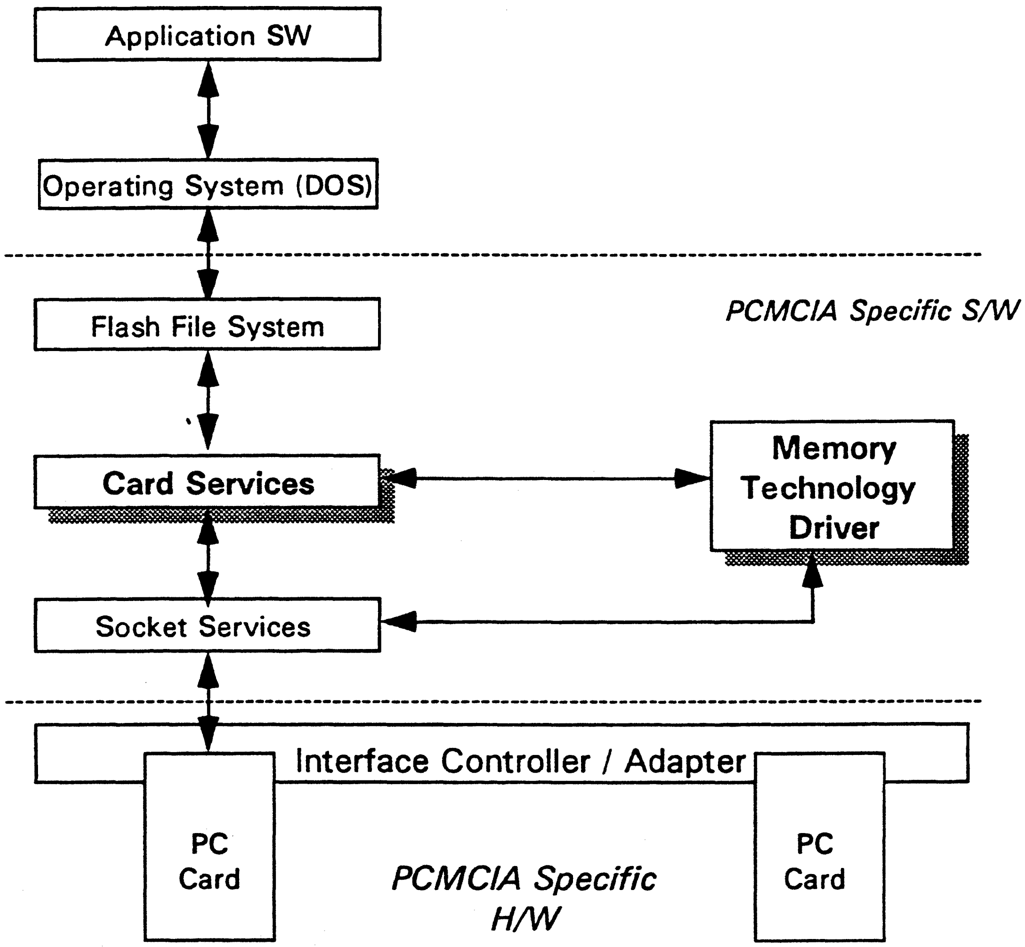

Figure 7.4 PCMCIA Layers: Focus on Card Services and MTDs

Card Services is a system resource manager that performs the following functions:

- Card Services allows higher-level software to work with various Interface Controller/Adapters. In other words, end users could use the same application software contained on a PC Memory Card in their desktop computer as on their notebook PC even if those two systems did not use identical PCMCIA hardware.

- Card Services is responsible for taking requests from multiple *clients* (higher level software) and servicing these requests. The Card Services interface is structured in a *client/server* model. Application programs, device drivers and utility programs are the clients requesting services. A Card Services implementation is the server fulfilling the request of the client. Socket Services arbitrates available system resources thereby eliminating potential hardware conflicts from multiple software clients.

Figure 7.5, "Card Services Software Architecture" shows the Card Services software architecture as depicted in the PCMCIA Card Services Specification. Again, the details of the diagram will be discussed in Volume II of this Training Manual.

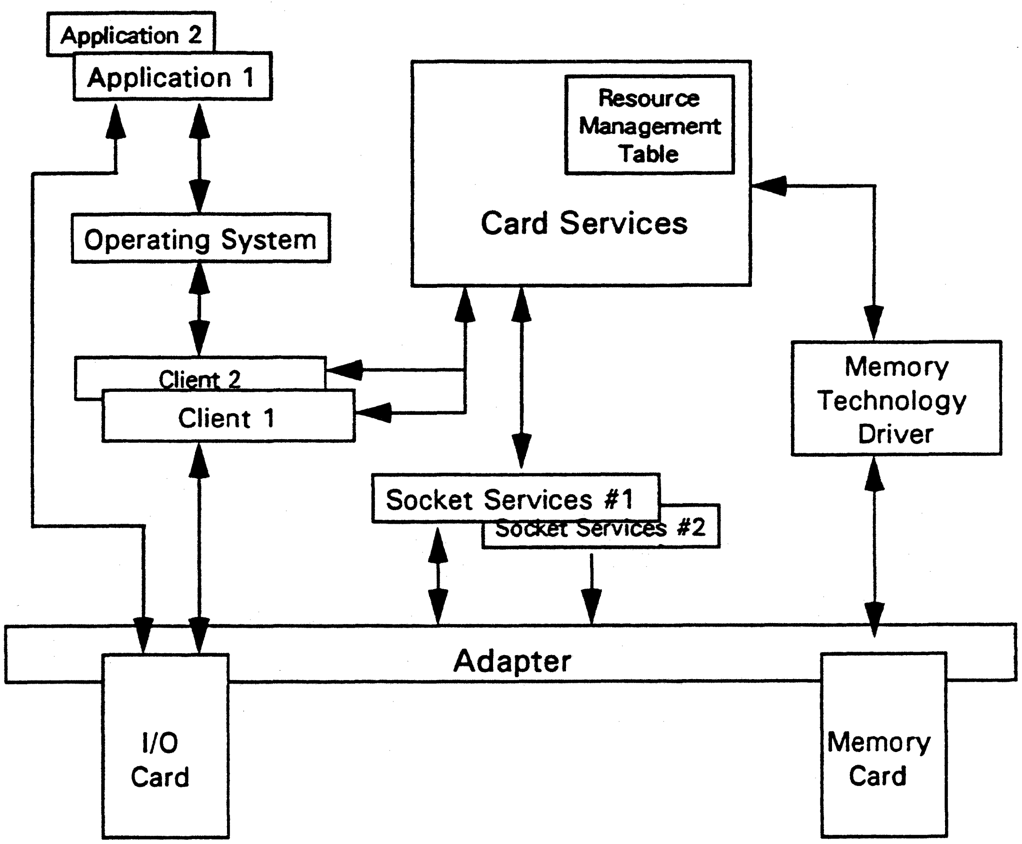

Figure 7.5 Card Services Software Architecture

Card Services minimizes the end user's involvement and skill needed to use various PC Cards, and provides transparent access to various media technologies used in PC Memory Cards via Memory Technology Drivers. Another goal of Card Services is to provide the end user of PC Cards with system specific interoperability. Also, Card Services allows the user to perform *hot-insertion*, which is the insertion of a PC Card into a PCMCIA socket while the socket is receiving system power. Hot-insertion of peripheral hardware is like changing the spark plugs in a car with the engine running. Since turning the power off or rebooting a system causes a loss of data stored in the system memory, Card Services performs *dynamic reallocation* of system resources without rebooting.

For example, a PC user, in the midst of working on a presentation, realized the need for a file that is in a co-worker's PC in an other city. This PC user has a modem but has not install it into his PC. In order to add a modem to his PC so he can call up his co-worker to down-load the needed file, the user has to:

Instructions for modem installation in an IBM PC without Card Services:

1. Save all files being worked on, exit the application program, and turn off the IBM compatible PC.
2. Remove the enclosure from the PC.
3. Discharge any static electricity from himself.
4. Install the modem Interface board into the expansion bus of the PC.

5. After careful consultation of the modem User's Guide that restricts the usable jumper position based on the hardware configuration of your PC, set the Interrupt setting on the modem Interface board.

   Warning: Failure to perform step 5 may result in hardware and/or software conflict that could cause system crashes.

6. Reinstall the PC enclosure.
7. Connect the modem to the telephone line.
8. Turn on the PC.
9. Insert the modem installation disk into the PC. Run the setup program to install the modem device driver to your CONFIG.SYS file.
10. Turn off or reboot the PC to reconfigure the system.
11. Modem is ready for use.

The above steps illustrate that certain skills and understanding of the PC fundamental such as Interrupt setting, hardware configuration, and bootup configuration are required by the user to add a modem to a PC. For most PC users, those skills and understandings are beyond the scope of their interest.

On the other hand, the same user with a PCMCIA compliant PC and Card Services, installing a modem is as simple as installing the modem device driver (a one time installation) and inserting the modem PC Card into the PCMCIA socket.

Instructions for modem installation in a PCMCIA compliant PC with Card Services:

1. Insert the modem installation disk into the PC. Run the setup program to install the modem device driver to your PC.
2. Save all files being worked on, exit the application program, turn off or reboot the PC to reconfigure the system.
3. Insert the modem PC Card into the computer and connect the modem to the telephone line.
4. Card Services configures the modem.
5. Modem is ready for use.

Steps 1 and 2 are performed one time only to give Card Services the location of the modem device drivers. After this initial software installation, the user can install the modem PC Card into the PCMCIA socket as needed without rebooting. This means that for a PCMCIA compliant PC with only one PCMCIA socket, the user can use a modem to down-load a file from another city, remove the modem PC Card, install a network PC Card into the same socket, call up a Local Area Network (LAN), and down-load another file into the PC. Both files can be down-loaded without turning the PC off or severely interrupting the user's work in process. Card Services allows the PC user to add PC peripherals without knowing the technical details of the PC and without wasting valuable time. This dynamic reallocation of system resources without rebooting is the key feature of Card Services that promotes user friendliness.

7.4.1 Card Service's Benefits for the End User

Two primary benefits of Card Services are the transparent access of PC Memory Cards via Memory Technology Drivers and the support for hot insertion of PC Cards without rebooting the PC. Card Services also:

- Provides a single device driver for supporting multiple PC Cards.
- Gives the system the ability to use multiple PC Cards in one socket.
- Supports multiple PC Cards/Sockets without hardware or software conflict.
- Supports different device drivers sharing a single PC Card.

  For example, a Flash card could be partitioned to have both a File Allocation Table format and Flash File System format on the same card, requiring two device drivers.

- Allows the system to use common default PC Card access functions. This minimizes the software support required to interact with both memory and I/O cards.
- Increases the performance of the system when unnecessary software notification is avoided.

7.4.2 Card Services Restrictions

There can be only one Card Services implementation per system. Although Card Services can be implemented by any Operating System, Card Services is invoked in a microprocessor and Operating System dependent manner. For example, Card Services from an IBM PC running DOS is different from the same IBM PC that uses OS/2 as its operating system. In other words, if the user deletes the DOS operating system from his hard disk or Resident Flash Array and installs the OS/2 operating system, Card Services will no longer function. In addition, Card Services 2.0 requires Socket Services 2.0 to function.

7.5 Memory Technology Drivers (MTDs)

A Memory Technology Driver's only purpose is to perform the basic memory Read/Write/Copy/Erase functions for Card Services. MTD requires Card Services to function. Thus, it is not a stand alone device driver. Memory Technology Drivers can be installed at boot time or it can be called into service as required. Card Services uses the MTD interface to access the MTDs. The MTDs use the *MTD Helper Routines* and the *Media Access Table Functions* to access the PC Card. MTDs register with Card Services like any client by using the *RegisterClient* request.

7.5.1 The Need for Memory Technology Drivers

Because different Flash manufacturers are producing Flash devices that require vendor specific programming algorithms, different Memory Technology Drivers are required for different Flash memory devices. A different MTD might be required for different capacity of Flash cards even if a family of Flash memory cards all use the same Flash devices. MTD written specifically for cards of the same part number will tend to be a more robust MTD then one that is written for a family of cards. This card specific MTD will also be smaller in size.

7.5.2 MTD Installation

MTD driver should be installed after the Card Services driver. In an IBM PC, MTDs are installed in the CONFIG.SYS file. MTDs can be distributed by many different methods. They can be embedded in the host system or within Card Services. They can be shipped on a diskette with the PC Card or the host system. PCMCIA is evaluating a method to allow MTDs to be shipped on the PC Card itself and to allow MTDs to be self loaded into the host system.

7.6 The Flash File System

The Flash File System (FFS) is a file system that can be installed in addition to the DOS File Allocation Table (FAT) file system. FFS is not an operating system and it can not function without DOS. The purpose of a Flash File System is to allow an end user to use a Flash memory card instead of a floppy disk in his PC without knowing that the Flash card's programming requirement is different from that of the floppy disk. Figure 7.6, "PCMCIA Layers: Focus on the Flash File System", shows how FFS fits into the hierarchy of the PCMCIA elements in an IBM Compatible PC.

7.6.1 Need for the Flash File System

During the course of a DOS file access, the File Allocation Table (FAT) is constantly updated as a file grows, shrinks, is created or deleted. The special requirements of the Flash memory technology such as Erase before Write of a block of data and a finite number of Erase/Write endurance cycles do not fit well with this characteristic of the DOS FAT file system. If the DOS FAT file system is imposed upon the Flash media, the constant updating of the FAT table will severely decrease the performance of the system and the useful life of the Flash media. The Flash File System changes the way DOS interacts with the Flash media without effecting the system performance and the Flash media's useful life.

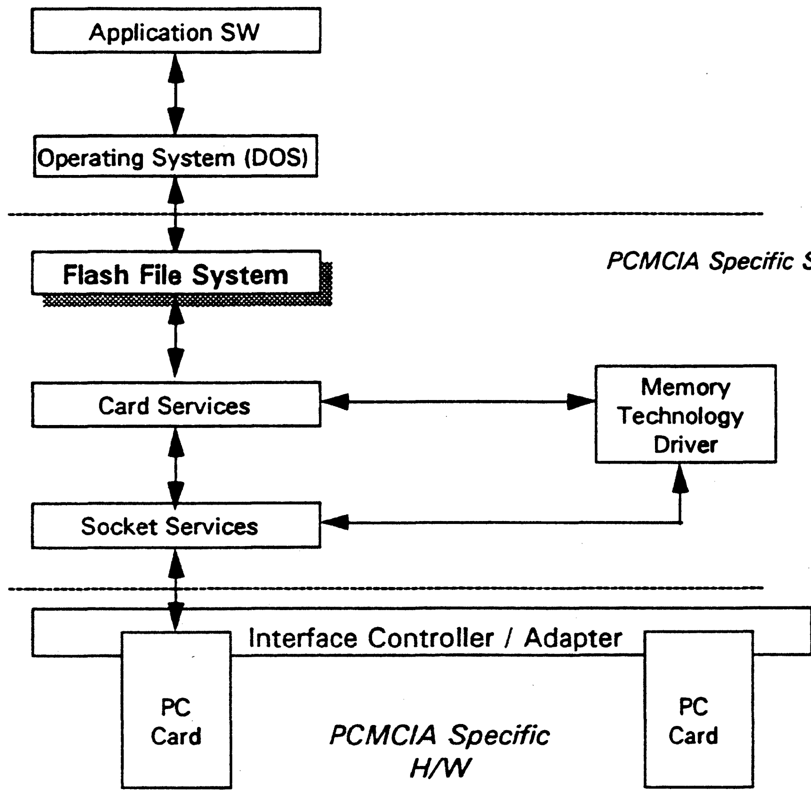

Figure 7.6 The PCMCIA Layers: Focus on the Flash File System

7.6.2 Benefits of the Flash File System

In a Flash File System, the directory and file control structures are evenly distributed among the data so that the updating of the FAT table does not result in constantly rewriting any fixed area of the media. The erasure of data blocks within the media is also evenly distributed to minimize the wear of the media. The Flash File System makes the Flash memory card appear as a floppy disk to the end user and provides the flexibility that the user expects from the traditional magnetic rotating media.

7.6.3 Flash File Systems from Different Suppliers

There are three companies marketing FFS today. Unfortunately, the three Flash File Systems are not compatible with each other and files stored in one FFS format are not accessible by the other two Flash File Systems. The three Flash File Systems are: Microsoft Flash memory File System, SCM-FFS by SCM Microsystem GmbH, and True FFS by M-Systems.

Chapter 8. Accessing PCMCIA Flash memory Cards

This final chapter will reiterate the topics introduced in the preceding chapters including an example of how an AMD Flash memory Card could be accessed in a Personal Computer prior to the implementation of Card Services and Socket Services. In the following example, we will discuss how other personal computers could read files from the AMD Flash memory Card using Microsoft's FFS 1.0 and a Databook TMB200 Thin Card Drive. We will compare how the AMD Flash memory Card is accessed via Card Services and Socket Services in the next volume of this Training Manual.

8.1 Hardware and Software needed to access a Flash memory Card

Obviously, a PCMCIA Compliant System with a 68-pin socket is the hardware needed before a user could write files to any Flash memory card. In this example, a Databook TMB200 Card Drive installed in an IBM PC is the PCMCIA Compliant System with a Type II 68-pin socket. The TMB200 is equivalent to the disk drive in a desk top PC and the Flash memory card is equivalent to the floppy disk. As in a desk top PC where a disk drive is assigned a drive letter, the TMB200 "disk drive" is Drive D in this IBM PC. When a user writes files to or reads files from drive D in this PC, the operating system activates a special set of software device drivers to control drive D. Please refer to Appendix B, "The IBM Compatible Personal Computer", for explanations on operating system and device drivers.

Depending on the system configuration of the user's PC, the TMB200 does not always appear as drive D in the system. For example, a user is using MSDOS 3.3, an 80 Mbytes hard disk, and a network connection in his PC. MSDOS 3.3 requires that the hard disk be partitioned into drive C, D, and E. If the network driver is loaded before the Databook TMB200 device drivers, the network will appear as drive F, and the TMB200 will be drive G. But if the Databook drivers are loaded before the network device driver, the Databook TMB200 will appear as drive F, and the network as drive G.

A file called *PCCARD.SYS* is the device driver in the IBM PC that controls the TMB200's PCMCIA socket. This device driver supports programming algorithms for Flash memory cards, OTPROM cards, and SRAM memory cards. PCCARD.SYS also supports Microsoft's Flash File System. Unless the user is using a Flash memory card that is supported by PCCARD.SYS, the Databook TMB200 will not be able to program this card. However, the TMB200 may be able to read files from this Flash memory card if the card has one of the two proper data file formats, the DOS File Allocation Table (FAT) or the Microsoft Flash File System.

When the IBM PC is first powered up, DOS carries out a group of commands that loads the installable device drivers. The file which contains these commands is called CONFIG.SYS. Since PCCARD.SYS is a device driver, it needs to be included in the CONFIG.SYS file that is located in the root directory of the boot-up disk drive. The command *"DEVICE= "* gives DOS the location and the name of the device driver. The DOS commands needed in the IBM's CONFIG.SYS file are:

DEVICE= C:\DATABOOK\PCCARD.SYS /D:310
DEVICE= C:\DATABOOK\PCFLASH.SYS
DEVICE= C:DATABOOK\IFS.SYS
DEVICE= C:DATABOOK\FEFS.SYS

The DOS command *"/D:310"* assigns I/O address 310 to the TMB200 Card Drive. The device drivers associated with Microsoft's Flash File System FFS 1.0 are: IFS.SYS, and FEFS.SYS. PCFLASH.SYS is the TMB200's device driver for the Flash File System.

8.2 Preparing a Flash memory Card for Data Storage

Like all new floppy disks, Flash memory cards need to be "formatted" before they are ready to store files. The "formatting" of a Flash memory card includes the "Erasing", "Initializing", and "Formatting" of the card. A utility program is used to "format" AMD's Flash memory card. The utility programs are: TCERASE.EXE, TCINIT.EXE, TCFORMAT.EXE, and TCXCOPY.EXE. The following DOS commands used to call up these utility programs were put together based on instructions from the DATABOOK Users' Manual for software release 2.11. Release 2.11 and later software are needed to program the AMD Flash Memory cards with 10K endurance cycles.

The DOS command used at the DOS prompt "C:\> " to erase AMD's 1 Megabyte Flash card (AmC001FLKA) is:

TCERASE -TYPE FLASH -SIZE 0x100000 D:

The DOS command to initialize this 1 Mbyte card is:

TCINIT -CONFIG AMD1M.CFG -CARD AMD\C001FLKA D:

The DOS program codes " *-CONFIG AMD1M.CFG* " call up the file named *AMD1M.CFG.* This file contains the card specific data to be recorded into the AMD card's Card Information Structure, and to store the "FILE-SYSTEM: DOS" <u>or</u> "FILE-SYSTEM: FlaSh" data on the CIS. The words "FILE-SYSTEM: DOS" tells the system the card has a DOS-FAT file structure whereas "FILE-SYSTEM: FlaSh" indicates the card has a Microsoft Flash File System 1.0 structure. The user need to choose which file structure to use with the card by changing the "FILE-SYSTEM: xxx" line in the AMD1M.CFG file. Finally, "*-CARD AMD\C001FLK*" tells PCCARD.SYS the card inserted into Drive D is an AMD 1 Mbyte Flash memory Card and the card's part number is AmC001FLKA.

8.3 Choosing a File Structure for the Flash memory Card

After the card has been erased and initialized, the user needs to decide what data file format he or she wants in this Flash memory card. As mentioned before, the TMB200 supports either the DOS FAT file format or the Microsoft FFS 1.0 Format.

8.3.1 Flash memory Card with a DOS FAT File Format

There are two ways to format the card with the DOS FAT file format. The first way is to use the TCFORMAT utility program by entering the DOS command:

TCFORMAT -CARD AMD/C001FLKA D:

The second method is to copy a DOS file with a file named "FLENAME.EXT" into the Flash memory card by entering the DOS command:

TCXCOPY FILENAME.EXT D:

If the user has formatted the card with the DOS FAT file data structure, the user will always need to use the TCXCOPY.EXE utility program at the DOS prompt to "write" files into this Flash memory card. This means that the application programs will not be able to write files to this card directly. In order to write a file to the Flash card, the user will first need to save the file to the hard disk or other mass storage media, then perform a TCXCOPY to transfer the file from the hard disk to the Flash card.

The file deletion function is not supported by PCCARD.SYS in a DOS FAT file format environment for Flash memory Cards. This means that in order to update a file, a user will have to rename the updated file before using the TCXCOPY utility to transfer the file to the card.

8.3.2 Flash memory Card with a FFS File Format

Alternately, the user can format the Flash card with Microsoft's Flash File System (FFS) by entering the DOS command:

TCFORMAT -CARD AMD/C001FLKA D:

Astute readers will notice that this is the same command used in formatting the Flash card with a DOS-FAT file format. The TCFORMAT utility program is used for both the FFS and the DOS-FAT file format. The actual file structure programmed into the card is based on the "FILE-SYSTEM: xxx" line in the AMD1M.CFG file. If xxx=DOS, then the DOS-FAT file structure will be used. If xxx=FlaSh, the Microsoft Flash File System 1.0 format will be used. Once the card has been formatted with the FFS file format, the card acts like a floppy disk. Application programs such as DBase, Lotus 123, Word Perfect, and Word for Windows also treat a FFS formatted card as a floppy disk. For example, the user can save files to this card in the "Word for Windows" application program by calling up the pull-down manual and selecting the "File - File Save As...- Drives: d:" commands. By selecting Drive D, the BIOS and the operating system will call up the necessary device drivers needed to program the file to the AMD Flash memory Card. The file deletion function is supported by the Flash File System.

8.4 Summary

Portable PC users expect to be able to treat their Flash cards like a floppy; therefore, DOS FAT file formatted Flash cards are not an ideal solution but they are useful for "Read only" applications or *Write Once Read Many* (WORM) applications. On the other hand, when a DOS FAT file format is programmed into a memory card, this card can be read by any other DOS system with a PCMCIA socket. If the card was programmed using Microsoft's Flash File System, then this card can only be read by another DOS machine with Microsoft's Flash File System installed. "Formatting" and Writing files to this card by the other DOS machine is only possible if that DOS system contains the proper software drivers and hardware for this card.

Without Card Services, the use of a Flash memory card today is not as simple as the use of a floppy disk. In the near future, Card Services will provide interoperability for Flash memory cards by accessing the Flash card via the card specific Memory Technology Driver. Card Services and Memory Technology Drivers will help Flash memory cards to be as easy to use as floppy disks in a PCMCIA compliant system.

Preview of the PCMCIA Training Manual Volume II

The PCMCIA Training Manual Volume II will give many of the technical details of PCMCIA. The material in this volume of the manual covers the topics that were introduced in Part II of the manual in much greater detail.

Some of the topics that will be covered are:

1. "PCMCIA Hardware", gives the details of the PCMCIA Hardware including the signal descriptions of the card, the controller, and the interface. This chapter also describes some design issues.
2. "PCMCIA Firmware: Socket Services", explains how this powerful PCMCIA tool serves a vital need in the connection of a PCMCIA card to the PC system.
3. "PCMCIA Software: Metaformat", describes the details of card specific data formats known as the Metaformat, and the software that contains card specific information known as Card Information Structure (CIS) and tuples.
4. "PCMCIA Software: Card Services and Memory Technology Drivers", explains how Card Services is used as the primary maintenance resource manager for PCMCIA cards and how MTDs, the slave to the Card Services master, is used to access memory on a PC Card.
5. "PCMCIA Software: The Flash File System (FFS)", describes how the FFS is used as a link from the world of DOS to the world of the PCMCIA card in a way similar to how a PC connects to a Local Area Network (LAN).

Some of the possible topics to be included in Volume II are:

- "X86 Microprocessor Fundamentals Important in PCMCIA", describes some of the fundamental concepts of the X86 microprocessor family that are especially important when interfacing with PC Cards including the internal and external bus width of the different microprocessors, registers, real mode, protected mode, and virtual machines.
- "ExCA ™ : Exchangeable Card Architecture", objectively addresses the myth of Intel's approach to combining parts of the PCMCIA Hardware, Socket Services, and Card Services.
- "XIP: eXecute In Place", explains how XIP gives the PC the ability to boot from a Flash memory PC Card by having the contents of the Flash memory mapped directly onto system memory allowing the PC to run the system directly from the card.
- "PCMCIA Working Groups", gives a list of the working groups and the status of topics on which they are working.
- "PCMCIA Auto Indexing Mass Storage (AIMS 1.0)", gives details of the AIMS 1.0 specification.
- "PCMCIA Advanced Technology Attachment (ATA 1.0)", gives detail of the ATA 1.0 specification.

The specific possible topics to be covered in Volume II will be based on the comments the authors receive regarding the effectiveness of Volume I and the interest readers have in the subjects listed above.

Appendix A: Solid State versus Rotating Mass Storage

This appendix provides an analysis of the relative strengths and weaknesses of Rotating Magnetic Storage Media, Flash memory PC Cards, and Flash memory Based *Integrated Device Electronic* (IDE) Drives in the portable form factors. Appendix A then summarizes the long-term value proposition of each approach to mass storage. The result of this comparison indicates there is a window of opportunity for Flash memory PC Cards in mobile applications that require relatively "small" amounts of storage capacity.

A.1 Analysis of Rotating Magnetic Storage Media

Using rotating magnetic media in portable devices for mass storage has four distinct advantages relative to Flash memory PC Cards.

1. Existing system-level software for IBM Compatible PCs, the Disk Operating System (DOS), can be used to operate the disk without additional software support. The electrical system interface (i.e., IDE) is well defined and DOS was written for the rotating magnetic drive.

2. Disk drive manufacturers have been focusing their R&D dollars on making these devices suitable for the mobile environment. For example, H.P.'s Kittyhawk 1.3 inch disk drive can withstand a drop from the height of three feet, as long as the disk drive lands in an upright position. In the past, disk drive manufacturers used most of their R&D money for cost reduction of the 2.5 inch and larger disk drives.

3. Some of the smaller disk drives are able to fit in the largest of the Memory PC Card packages, namely the PCMCIA Type III PC Card which is 10.5 mm thick.

4. Disk drives are priced lower on a dollar per Mbyte basis relative to Flash memory. In 1992, MiniStor introduced a 1.8 inch disk drive for less than $6 per Megabyte at a 64 Mbyte density. H.P.'s Kittyhawk, meanwhile, is a 1.3 inch disk drive selling for less than $12.50 per Mbyte in the 21.4 Mbyte density. Obviously, these products provide cost effective mass storage relative to Flash memory, which is still in the range of $50/Mbyte using the highest density Flash memory devices available. However, the benefit of a lower cost/Mbyte is only a relevant measure as long as the end user needs all of the storage capacity that he or she is paying for. For example, if a user only need 20 Mbytes of storage, but the lowest density drive available is a 100 Mbyte disk drive, then the user is paying for 80 Mbytes of storage that he or she does not need. Thus, in this scenario, the user's effective dollars per Mbyte is five times the dollars per Mbyte that the disk offers if all 100 Mbytes were required.

The economics of disk drive production creates a cost floor, or minimum price, below which drives are not sold. This cost floor is composed of both silicon and mechanical content. The cost structure associated with disk drives has provided the leverage that disk drive makers have exploited by increasing the amount of storage capacity on a fixed cost platform to lower the price per Mbyte of storage. In other words, the incremental cost of additional storage capacity on a given form factor is a relatively small proportion of the overall cost of the drive.

Rotating mass storage media also has at lease five disadvantages relative to Flash memory when used in a mobile application.

1. A disk drive based system relies on a *two tier memory hierarchy* (DRAM + Disk) for operation. Code is not executable directly from the drive. A Resident Flash Array and memory card architecture, similar to the one described in Chapter 3, "The Portable computer User", eliminates this redundancy. In this case, the system DRAM is used only for the temporary storage and display of working files, thus less DRAM capacity is needed.

2. There is a significant time users spend waiting for their computers during power-up, or when the system resumes from a sleep mode. This is caused by the spin-up and spin-down of the disk drive required to transfer executable code to DRAM and save power, respectively. Executable code stored in the Flash array provides for Instant Access capability, thereby eliminating any delay seen by the user in these circumstances.

3. Even though some disk drives are available in the PCMCIA Type III PC Card package, and are more portable than their larger sized predecessors, they are still not as portable as Flash memory PC Cards which are available in PCMCIA Type I packages that are 3.3 mm thick. The large size of the Type III PC Card means that it occupies the space of two horizontally oriented PCMCIA slots. This means a system would require three PCMCIA slots in order to use any other add-on card while using the disk. This limits the flexibility of a portable's design.

4. Portable disk drives are not as shock resistant as solid state Flash memory PC Cards. Even the Kittyhawk will not withstand a drop while the disk is running if the device lands anywhere other than on its bottom. Consider how many times that your calculator lands exactly that way?

5. A disk drive requires at least 2.5 Watts of power during spin up and 1.5 Watts while spinning. The power consumption of any rotating disk drive will always be orders of magnitude greater than that of the Flash memory Card's power needs.

In conclusion, rotating magnetic media has a long term value proposition in the larger form factor portables such as notebooks and portions of pen-based and sub-notebook machines. These larger machines tend to have the space available for larger and more powerful battery packs to satisfy the disk drive's power budget. There is more space available in the system box for the larger Type III PC Card and enough board space for the "redundant" system DRAM. In addition, the larger portables are more likely to be used on solid surfaces or desks while out of the office. Therefore, they will not be subjected to the same requirements for ruggedness as the smaller form factor machines.

A.2 Analysis of Flash memory PC Cards

The benefits of Flash memory Cards lie in the areas where disk drives have their weaknesses. This mass storage alternative is particularly suited to the portable environment. The solid state Flash memory card extends the operating battery life of mobile equipment because it consumes extremely low power relative to the rotating disk (i.e., milliwatts versus watts). In addition, this also allows for a decrease in system weight by using smaller and lighter batteries.

Flash memory PC Cards are extremely rugged because they have no moving parts that may be damaged if dropped. In the mobile environment, this provides system designs with an added level of reliability not achievable with rotating media. Flash memory PC Cards are also available in the small, PCMCIA Type I package.

Flash memory cards provide the lowest cost alternative for mass storage applications where relatively small amounts of capacity are required. When the end user has a small dollar budget for his or her storage capacity, Flash memory PC Cards can be purchased in Mbyte increments. Thus, the user pays for only the amount of memory that he or she requires. Disk drives on the other hand are typically not available at any density below a certain cost floor. This cost floor is typically between $150 and $200. If this amount exceeds the end users' budget, the drive may not be a viable alternative for their application.

Some of the disadvantages of the Flash memory cards are actually being addressed by the PCMCIA Software Committee and are expected to be resolved by November, 1992. These issues include the development of a new industry wide infrastructure in system-level software. This software is required to drive the interface from current operating systems, such as DOS, to the Memory PC Cards. It also allows a user to operate a Memory PC Card-based system just like a disk-based system. These software issues are discussed in detail earlier in this manual and are referred to as Socket Services, Card Services, and Memory Technology Drivers. Once this software is standardized, the full benefits of the Memory PC Cards may be realized. These benefits are discussed in Volume I and Volume II of this manual.

One of the other disadvantages of Flash memory PC Cards is its slower write performance relative to disk drives. However, this is expected to be addressed in the same manner as it has been with disk drives. Thus, *write caches* are expected to be used in order to buffer the user from noticing any write performance degradation while using a Flash memory PC Card.

In conclusion, Flash memory PC Cards have their long term value proposition in the smaller form factor machines. These include units from the hand-held, sub-notebook, and pen tablet portable PC categories. These form factor machines typically have smaller and lighter battery packs and do not have the space for more than two PCMCIA slots. The usage model of these devices indicates that they will not be the primary computational unit of the user, but an adjunct. In this sense, the user may only need to carry sufficient storage capacity for the work they intend to perform while away from the office. These files will be transferred between the portable and the end user's primary computer via a Flash memory PC Card. Thus, an end user will be able to use a Memory PC Card with relatively smaller storage capacity than his or her desktop unit. In addition, these end users are more likely to use their portables in truly mobile environments where ruggedness is valued at a premium. Solid state Memory PC Cards provide the highest level of reliability in these applications.

A.3 Analysis of Flash Based IDE Drives

This product uses the same electrical interface as the Integrated Device Electronic (IDE) disk drive but uses Flash memory as the storage media. Therefore, the Flash-based IDE drives share some of the pros and cons of both the rotating media and the Flash memory PC Cards. Like the disk drives, the IDE Flash drives can use the existing system level software infrastructure (i.e., DOS). And like the Flash PC Cards, this product consumes very little power relative to the disk drive. In addition, the solid state nature of the IDE Flash drive makes it more rugged and portable than rotating media.

Some of the disadvantages of the Flash-based IDE drives are similar to those of the rotating disk. However, there are also some disadvantages that are unique to this product. Like the disk drives, the Flash-based IDE drives require a two tiered memory hierarchy for operation of the system. A disadvantage that is unique to this device is that it must fit the Flash memory media into a Disk (rotating) Operating System. This is accomplished with the use of a central controller chip in addition to the Flash memory devices. The IDE controller performs the translation of DOS commands that will drive the Flash memory technology. The cost structure of this approach is greater than that of a Flash memory PC Card by about 50% to 60%. These items taken together tend to limit the implementation of this type of product to the interim period before PCMCIA Software support is implemented.

Thus, the Flash-based IDE has a limited window of opportunity. Once Card Services are implemented, the benefits of the Flash memory PC Card can be fully utilized. This includes the lower cost per Mbyte relative to the Flash-based IDE drives and the direct execution capability of the Flash memory PC Cards. Flash-based IDE drives are not expected to be a cost effective product in the long run.

A.4 Conclusion: The Window of Opportunity for Flash PC Cards

The two long term alternatives to mass storage in the mobile environment are expected to be the rotating magnetic media (i.e., disk drives) and Flash memory PC Cards. The rotating disk drives will dominate the larger form factor applications. In this case, the end user's dollar budget for storage capacity is in line with the minimum floor cost and higher density available with disk drives.

Flash memory PC Cards will dominate the smaller form factor applications that place a premium on ruggedness, small form factor packaging, light weight, and long battery life of the system. These machines will likely be used as advanced organizers, Personal Information Devices (PIDs) with application specific software such as Multiple Listing Service data bases for real estate agents, or in pen based devices used by hospital nurses, the police, delivery drivers, and inventory control clerks, and as removable media in the sub-notebook form factor. The expansion of PIDs into the truly mobile environments (i.e., not just transporting a notebook from one desk location to another) will be enabled by the Flash memory PC Card. Ruggedness is expected to be one of the most important value propositions for truly mobile PCs. It is expected that this value proposition will be so strong, that it will allow a moderate price premium on a dollar per Mbyte basis relative to disk drives. In addition, these applications are expected to require moderate amounts of storage capacity relative to a full featured notebook or desktop computer. For example, handhelds and application specific devices are expected to require between 10 to 100 Mbytes of storage capacity.

Figure A1, "Media Storage Trends for Portable PCs", shows the window of opportunity for Flash memory versus disk drives over time. Typically, the larger form factor notebooks will be configured with traditional mechanical disk solutions. The smaller form factor hand-helds will be configured with Flash PC Cards. The pen tablet and sub-notebook form factors represent an area of overlap between the rotating disk and the Flash memory PC Card solutions. The end user requirements of the equipment will likely be the deciding factor in which technology is implemented. If a truly mobile environment is targeted, or an environment that requires long battery life, then the Flash memory PC Card will dominate. However, if the end users are price sensitive, or need larger amounts of storage capacity, then the rotating disk drives will dominate. The disk drives dominate the higher storage density portion of the graph, while the PC Cards dominate the low density applications. Figure A1 shows the amount of mass storage (in Megabytes) a user can purchase with the $150 - $200 from the 1992 to 1997 time frame. For example, in 1992, a user can purchase a 64 Megabyte 1.8 inch hard disk for $150 to $200 or a 1 Megabyte Flash memory card for the same price range. But by 1997, the same $150 to $200 can purchase a 100 Megabyte hard disk or a 67 Megabyte Flash memory card.

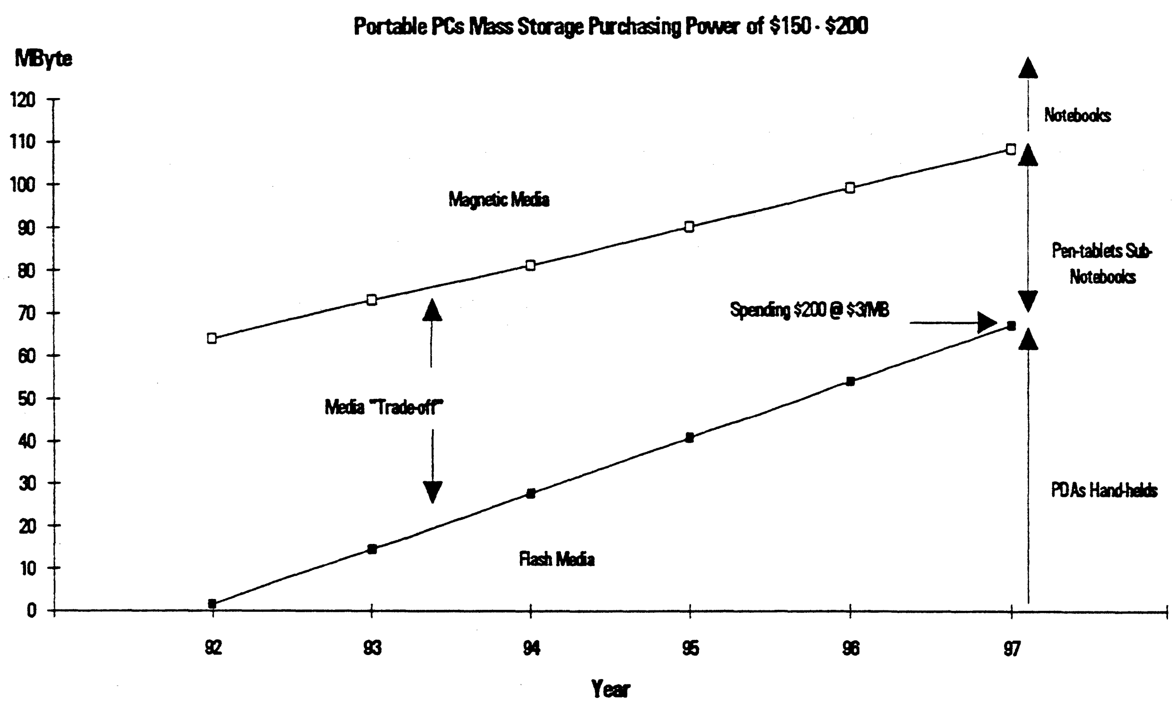

Figure A1 Media Storage Trends for Portable PCs

Figure A1 indicates the storage media of choice for different form factor machines and summarizes the conclusions previously discussed.

- Notebook class products are likely to incorporate magnetic media as the preferred storage device.
- Personal Digital Assistants and hand-held computing products are likely to use Flash media as the preferred storage device.
- Users are likely to trade-off the benefits of magnetic and Flash media depending on the work environment that pen tablets and sub-notebooks are used.
- The bottom diagonal line represents the price umbrella under which Flash memory PC Cards are likely to offer a superior value proposition relative to magnetic media. This price umbrella is expected to be in the range of $150 to $200. The amount of Flash media depends upon the actual dollar per Megabyte ratio available in 1997. This is illustrated in Figure A2.

Figure A2, "Flash Storage Trends ($/MB)", shows the envelope of dollar per Megabyte price projections for Flash Memory cards. The upper boundary depicts a scenario where the price of Flash cards is at $3/Megabyte in 1997, and a user can purchase a 67 Megabyte card for $200. The lower boundary shows that if the price of Flash cards is at $5/Megabyte in 1997, a user can still purchase a 30 Megabyte card for $150.

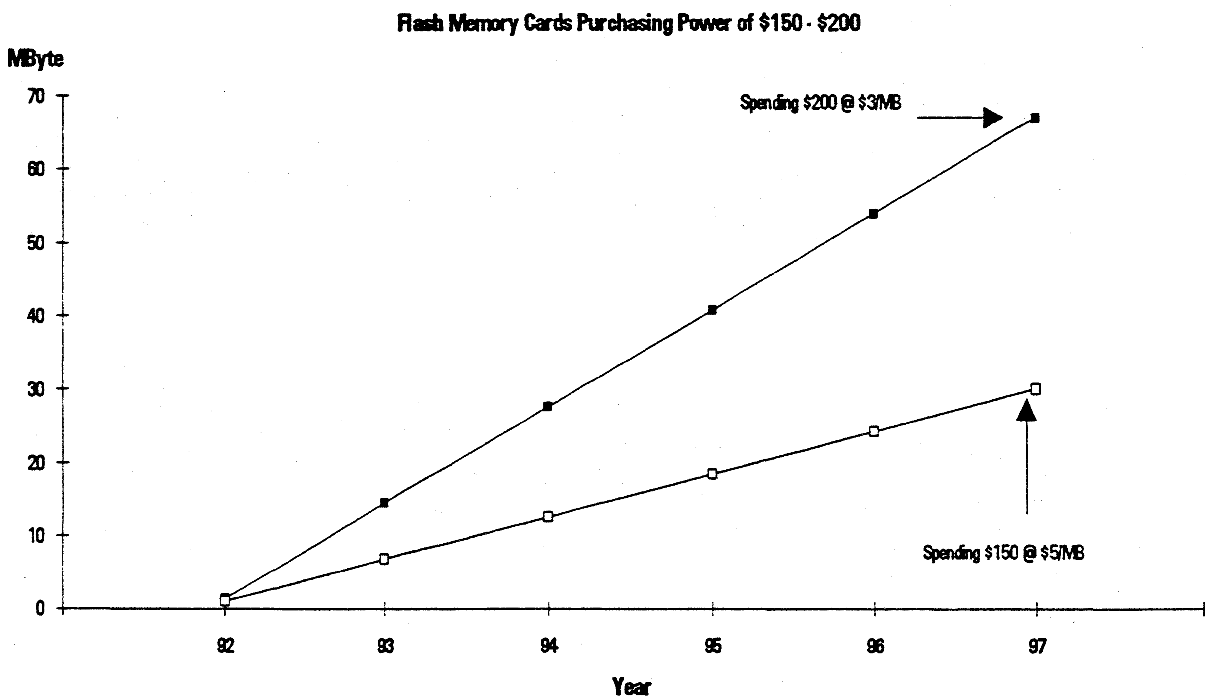

Figure A.2 Flash Storage Trends ($/MB)

Appendix B: The IBM Compatible Personal Computer

B.1 The History of the IBM Personal Computer

The IBM personal computer (IBM PC) was released in 1981. It was based on the Intel 8088 microprocessor, and its introduction legitimized the young personal computer industry. The IBM PC was not the first personal computer, but it was the machine that changed the PC from a plaything for the hobbyist into a serious business tool.

The original IBM PC is not powerful by today's standards. It was released with a total of 16 Kilobytes (Kbytes) of RAM, expandable to 64 Kbytes on the motherboard. The original 8088 microprocessor ran at 4.77 MHz, a speed now considered too slow for business or professional applications, and was capable of addressing a maximum of 1 Megabyte (Mbyte) of RAM. In addition, IBM chose the 8088, with 16-bit internal processing capability and an 8-bit external bus, instead of the 8086 microprocessor, with a 16-bit internal and external bus, so that the PC could be configured with inexpensive 8-bit peripherals (such as disk drives) and 8-bit microprocessor support chips.

Later versions of the IBM PC used more advanced X86 microprocessors. The 286 and later microprocessors, offered an new "mode" for accessing memory known as the "protected mode". The 8086 and 8088 microprocessors only used "real mode", which allowed 1 Mbyte of memory to be accessed directly. However, DOS breaks this 1 Mbyte of RAM into two components: low conventional memory from 0-640 Kbytes, and high conventional memory from 640 Kbytes to 1 Mbyte. The high conventional memory is reserved for system functions, such as system BIOS and DOS; thus, only 640 Kbytes of memory are accessible to the PC user in real mode using DOS. Protected mode allows the microprocessor to access larger amounts of memory directly, and it also allows *multi-tasking* on a PC. Unfortunately, DOS cannot take advantage of protected mode. The user must have an operating system such as Windows 3.0 (or higher) on top of DOS to take advantage of the multi-tasking capabilities that the protected mode offers. More information about the X86 microprocessor family is contained in the Volume 2's "X86 Microprocessor Fundamentals" section. The use of protected mode, and how it relates to PCMCIA memory cards is addressed in Volume 2., the "PCMCIA Firmware: Socket Services" chapter. The detail of the Real Mode and the Protected Mode of operation will be discussed in Volume 2 of this Training Manual.

In 1987, under competitive pressures from IBM Compatible manufacturers, IBM introduced the IBM PS/2 line. The PS/2 was a series of computers based on the 8086, 286, and 386 microprocessors. IBM abandoned the open bus architecture in favor of the proprietary Micro Channel Bus. Few IBM PC Compatible manufacturers have chosen to use the Micro Channel Bus because it requires a license to be purchased from IBM.

Since few companies adopted the Micro Channel Bus, the old open architecture bus was renamed the *Industry Standard Architecture* (ISA) bus for 16-bit data. Later, the IBM PC Compatible manufacturers adopted another open architecture bus known as the *Extended Industry Standard Architecture* (EISA) for 32-bit data. The EISA bus allows for an easy upgrade from the ISA bus architecture. One should note that the reference to IBM PC compatibles today typically refers to personal computers using an ISA bus.

B.2 Overview of the Structure of a PC

The personal computer can be thought of as a series of layers where the lowest layer is the physical hardware (the central processing unit (CPU), the disk drive, etc.) and the highest layer is the applications programs. (e.g., a word processing program, a spreadsheet program, etc.) Figure B.1, "The Layers of the Personal Computer", shows the layers of the PC. The following sections describe each layer of the PC from the lowest to the highest layer.

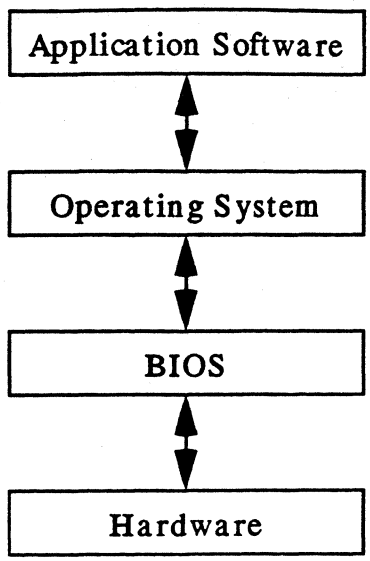

Figure B.1 The Layers of the Personal Computer

B.2.1 The PC Hardware

PC hardware is a collection of electronics and electro-mechanical components. Some components operate at the system level, and others operate at the peripheral level. System level components are integral to the PC's computing architecture. The CPU (Central Processing Unit) is an example of a system level component. System memory, chip-set logic, and VGA screen controller are other examples. System-level hardware in IBM PC, PC AT, PS/2, and IBM PC Compatible computers have CPUs that belong to the X86 microprocessor family.

Peripheral hardware, which connects to the system hardware, provides inputs and accepts outputs. Disk drives, keyboards, modems, mice, display terminals, printers, and a PCMCIA interface plus a PC Card are types of peripheral hardware.

B.2.1.1 System-Unit Hardware

The system-unit hardware is the brain of the computer. This hardware enables the computer to do all of the "thinking". System-unit hardware includes the microprocessor, memory, and various control chips. What the system-unit hardware does not include is access to the outside world, or communication with people - that task is handled by peripheral hardware.

The *Central Processing Unit* (CPU) is the part of the computer that examines, manipulates, and transfers data from one place to another in the computer. In effect, the CPU is the master controller for the computer. In large computers, the CPU may be comprised of many circuit boards containing dozens of chips. In PCs, however, the CPU essentially is contained in one chip, the microprocessor.

System memory refers to the chips in which the computer stores instructions and the data being used. Currently, volatile memory, or memory that does not retain its data when the power is turned off, is used for system memory. *Dynamic Random Access Memory* (DRAM) is usually used as system memory since it is less expensive than *Static Random Access Memory* (SRAM). However, some computers use SRAM as the cache memory. Since SRAM is much faster than DRAM in random accessing of data, cache memory helps to speed a computer's memory access time. One note, however, if a computer has cache memory, it must also have a *cache controller* to manage the cache memory.

Non-volatile memory, memory that retains data when the power is turned off, had not been used historically for system memory since it could not be programmed in system. However, the invention of Flash memory, for the first time, gives system users a cost effective, solid state, in-system programmable non-volatile memory. This is an incredible advantage for system users. In the future, computers may have Instant On capability since the system will boot directly from the system Flash memory, and the systems will run much faster since data will not have to be accessed from disk drives into RAM, but instead executed directly from system Flash memory.

Although the CPU supervises the activities of the PC, the CPU relies on a collection of support circuits to perform tasks. The CPU communicates with these support circuits by signaling them over electrical pathways called busses. Most computers have several busses. The bus through which the CPU coordinates computer activity is called the *control bus*. When the computer wants to access a memory location, it specifies the location via the *Address Bus*, and the data is transferred through the *data bus*. Many components are attached to the busses in a telephone party line-like arrangement. Part of the function of the control components is to ensure that the correct device is the recipient or sender of information transferred from place to place within the computer.

With numerous components connected to the three busses, successful computing requires a high degree of internal coordination among components. Control components, through the supervision of the CPU, provide this coordination. All the components in the system-unit hardware step in time to a system clock. At regular intervals, the system clock sends pulses that *synchronizes* the work cycles of the components.

While the *system clock* keeps the activities synchronized inside the computer, things can happen outside the computer at almost any time. Events such as your pressing a key on the keyboard or clicking the mouse have no way of falling in step with the smooth flow of the clock-controlled activity. Outside events happen in their own time frames, not in the time frame of the system clock. Hence, outside events are *asynchronous* to the system clock.

To service an outside event, the CPU must interrupt what it is doing and give immediate attention to the outside event, or interrupt. As soon as possible, the CPU executes a special block of instructions designed to handle the outside event that has occurred. A special control component, the *interrupt controller*, is tasked with detecting external events and informing the CPU that the current series of instructions must be put on hold. The interrupt controller then informs the CPU where the instructions that process the outside event can be found. The CPU can then begin executing the interrupt service routine to process the interrupt condition. When the outside event is handled, the CPU picks up where it left off and continues processing. In effect, the interrupt controller has taken an asynchronous outside event and made possible the handling of that event by a synchronous system hardware. When more than one event needs the attention of the CPU, the interrupt controller can determine the order in which the outside events are to be handled.

B.2.1.2 Peripheral Hardware

Peripheral hardware is hardware not immediately associated with the PC's system circuits; rather it is the equipment attached to the system-unit hardware. Peripheral hardware is often defined as the equipment that enables the computer to communicate with the outside world, such as a printer, keyboard, or screen. Peripheral hardware may also be some type of equipment controlled by the computer, such as laboratory equipment, or long-term storage devices such as a disk drive or optical disk. Because peripherals provide the computer with input or output, they are sometimes called *Input/Output (I/O) devices*.

B.2.2 The Basic Input/Output System (BIOS)

The logic that drives and controls the peripheral hardware is the Basic Input/Output System, or *BIOS*, which is a set of low-level, pre-programmed routines. The BIOS deals with the hardware specifics. BIOS routines are well-defined, binary-program segments that use the peripheral hardware's features and provide a uniform access to hardware functions. BIOS routines provide a standard interface between the hardware and higher level software such as DOS. This interface frees DOS from the need to know exactly how to use the hardware at a binary level.

BIOS routines typically lie in Erasable Programmable Read-Only Memory (EPROM), which is located at a predetermined address within the address range of the CPU. However, more and more frequently, BIOS routines are located in Flash memory. By locating the BIOS on Flash memory, the BIOS is still retained while the power of the machine is turned off , like EPROM and other non-volatile memories, but the BIOS program code can be changed in system with Flash memory, whereas with EPROM, the chip needs to be removed to change the BIOS. Reprogrammable BIOS allows a user friendly and cost effective upgrade of the system. After the BIOS is loaded during the boot process, the BIOS is stored in reserved locations of system memory. To protect the bootable portion of the BIOS, the Flash memory usually has a sector that cannot be erased in the system, known as a protected sector, so that the bootable portion of the code will not be inadvertently erased.

The BIOS programs of IBM Personal Computers, XTs, ATs, and PS/2s are copyrighted. PC compatible manufactures must create a BIOS that emulates IBM BIOS without actually using IBM's code. Companies that manufacture IBM PC compatible computers must create a BIOS emulation themselves or purchase a BIOS emulation from another company, such as Phoenix Technologies or Award Software.

B.2.3 The Operating System

B.2.3.1 Disk Operating System (DOS)

The Disk Operating System, or DOS, consists of two layers: the *DOS Kernel*, which is the core of DOS, and the command processor, known as *COMMAND.COM*.

The DOS Kernel is the hidden layer of DOS that is contained on the system boot disk. The hidden file is usually named IBMDOS.COM or MSDOS.SYS. The DOS kernel manages the file system, handles character input and output, oversees memory allocation, and performs other operating system tasks. At its lowest level, the DOS kernel calls on the service of the BIOS layer where the kernel uses the BIOS routines as building blocks to supply operating system service to programs. DOS contains a number of *device driver*, or software routines that interface with BIOS to access peripherals. In some cases, however, the user must install a device driver to interface between DOS and the BIOS to access a peripheral. This is usually the case with a mouse. One of the greatest features of DOS is that it allows the user to add device drivers as peripheral technology evolves. To access the services of the DOS kernel to copy a file or make a directory, the PC user must go through the next outer layer of the PC, COMMAND.COM.

DOS uses a hierarchical file system. The main directory of the file system is called the root directory. Below the root directory, the system can have sub-directories or files. Each directory or sub-directory may also have additional files.

Since DOS was designed for storing files on rotating magnetic media (disks), the DOS file system is stored using a file system designed for sequential storage on a flat circular media. The table of contents for a disk is known as the File Allocation Table (FAT), and it is the key to DOS's management of disks and files.

COMMAND.COM, the command processor program, is a layer of services that resides above the DOS kernel. COMMAND.COM provides a standard set of commands that gives the PC user access to file-management, device-management, configuration, and miscellaneous functions, such as maintaining and verifying the time and date. COMMAND.COM deals with user commands via a separate module from the kernel and is the kernel's command-interpreting shell. You can recognize this layer of a working PC by the *DOS prompt* " C:\> ".

B.2.3.2 The Windows Interface

Microsoft Windows is a graphical user interface that runs on top of the DOS operating system. The Windows environment is not required to run a PC, but it is becoming more common, and a cadre of applications programs have been developed around the Windows interface.

Windows 3.0 (and higher) can run DOS applications in protected mode. Thus, Windows 3.0 has a memory management system that gives the user the capability to run multiple applications at once, in addition to the graphical user interface. The full power of Windows is not realized, however, unless the PC being used has a minimum of 4M of RAM and a 386SX microprocessor.

B.2.4 The Application Programs

The *application program* layer is the highest level of the PC system. Applications programs include word processors, spreadsheets, databases, presentation programs, games, and other software.

B.3 Summary

This appendix discussed the history of the IBM Personal Computer and the structure of the PC.

The history of the personal computer is an interesting story. The first IBM PC was introduced in 1981, and the industry has experienced phenomenal growth ever since. IBM Compatible PC's entered the market in the early 1980s, and gave IBM tremendous competition in the PC market. In 1987, IBM made a major strategic change when it changed from its open system architecture (now known as the ISA bus) to a proprietary architecture. Most PCs today still use the ISA bus, and the PCMCIA standard was developed with the ISA bus architecture in mind.

The basic structure of the PC can be thought of as a series of layers. The layers, from lowest to highest, are the Hardware, the Basic Input/Output System, or BIOS, the Operating System, DOS, usually with Microsoft Windows, and the Applications Software.

Appendix C: PCMCIA PC Card Pin Assignments

Memory Only Card Interface (Always available at card insertion)						I/O and Memory Card Interface (Available only after card/socket are configured)				
Pin	Signal	I/O	Function	+/#	Notes	Pin	Signal	I/O	Function	+/#
1	GND		Ground			1	GND		Ground	
2	D3	I/O	Data bit 3			2	D3	I/O	Data bit 3	
3	D4	I/O	Data bit 4			3	D4	I/O	Data bit 4	
4	D5	I/O	Data bit 5			4	D5	I/O	Data bit 5	
5	D6	I/O	Data bit 6			5	D6	I/O	Data bit 6	
6	D7	I/O	Data bit 7			6	D7	I/O	Data bit 7	
7	CE1	I	Card Enable 1	#	3	7	CE1	I	Card Enable 1	#
8	A10	I	Address bit 10			8	A10	I	Address bit 10	
9	OE	I	Output Enable	#		9	OE	I	Output Enable	#
10	A11	I	Address bit 11			10	A11	I	Address bit 11	
11	A9	I	Address bit 9			11	A9	I	Address bit 9	
12	A8	I	Address bit 8			12	A8	I	Address bit 8	
13	A13	I	Address bit 13			13	A13	I	Address bit 13	
14	A14	I	Address bit 14			14	A14	I	Address bit 14	
15	WE/PGM	I	Write Enable	#		15	WE/PGM	I	Write Enable	#
16	RDY/BSY	O	Ready/Busy	+/#	2, 4	16	**IREQ**	**O**	**Interrupt Request**	+/#
17	Vcc					17	Vcc			
18	Vpp1		Programming Supply Voltage 1		2, 3	18	Vpp1		Programming Supply Voltage 1	
19	A16	I	Address bit 16			19	A16	I	Address bit 16	
20	A15	I	Address bit 15			20	A15	I	Address bit 15	
21	A12	I	Address bit 12			21	A12	I	Address bit 12	
22	A7	I	Address bit 7			22	A7	I	Address bit 7	
23	A6	I	Address bit 6			23	A6	I	Address bit 6	
24	A5	I	Address bit 5			24	A5	I	Address bit 5	
25	A4	I	Address bit 4			25	A4	I	Address bit 4	
26	A3	I	Address bit 3			26	A3	I	Address bit 3	
27	A2	I	Address bit 2			27	A2	I	Address bit 2	
28	A1	I	Address bit 1			28	A1	I	Address bit 1	
29	A0	I	Address bit 0			29	A0	I	Address bit 0	
30	D0	I/O	Data bit 0			30	D0	I/O	Data bit 0	
31	D1	I/O	Data bit 1			31	D1	I/O	Data bit 1	
32	D2	I/O	Data bit 2			32	D2	I/O	Data bit 2	
33	WP	O	Write Protect	+	2, 3	**33**	**IOIS 16**	**O**	**IO Port is 16 bit**	+
34	GND		Ground			34	GND		Ground	

Notes:

1. Wait and Reset are RFU (no connect) in Release 1.0. Both must be implemented in the system for Release 2.0 compliance.
2. Use of signal changes between memory only and I/O interface.
3. Signals must not be connected between cards.
4. Signals must not be connected between cards when I/O interface is supported.
5. Reset must not be connected between cards unless all cards are reset when any card has Vcc power removed.
6. In systems which switch Vcc individually to cards, no signal should be directly connected between cards other than ground.

Memory Only Card Interface (Always available at card insertion)						I/O and Memory Card Interface (Available only after card/socket are configured)				
Pin	Signal	I/O	Function	+/#	Notes	Pin	Signal	I/O	Function	+/#
35	GND		Ground			35	GND		Ground	
36	CD1	O	Card Detect 1	#	3	36	CD1	O	Card Detect 1	#
37	D11	I/O	Data bit 11			37	D11	I/O	Data bit 11	
38	D12	I/O	Data bit 12			38	D12	I/O	Data bit 12	
39	D13	I/O	Data bit 13			39	D13	I/O	Data bit 13	
40	D14	I/O	Data bit 14			40	D14	I/O	Data bit 14	
41	D15	I/O	Data bit 15			41	D15	I/O	Data bit 15	
42	CE2	I	Card Enable 2	#	3	42	CE2	I	Card Enable 2	#
43	RFSH	I	Refresh			43	RFSH	I	Refresh	
44	RFU		Reserved		2	44	**IORD**	**I**	**IO Read**	#
45	RFU		Reserved		2	45	**IOWR**	**I**	**IO Write**	#
46	A17	I	Address bit 17			46	A17	I	Address bit 17	
47	A18	I	Address bit 18			47	A18	I	Address bit 18	
48	A19	I	Address bit 19			48	A19	I	Address bit 19	
49	A20	I	Address bit 20			49	A20	I	Address bit 20	
50	A21	I	Address bit 21			50	A21	I	Address bit 21	
51	Vcc					51	Vcc			
52	Vpp2		Programming Supply Voltage 2		2, 3	52	Vpp2		Programming Supply Voltage 2	
53	A22	I	Address bit 22			53	A22	I	Address bit 22	
54	A23	I	Address bit 23			54	A23	I	Address bit 23	
55	A24	I	Address bit 24			55	A24	I	Address bit 24	
56	A25	I	Address bit 25			56	A25	I	Address bit 25	
57	RFU		Reserved			57	RFU		Reserved	
58	RESET	I	Card Reset	+	1, 5	58	RESET	I	Card Reset	+
59	WAIT	O	Extend bus cycle	#	1, 3	59	WAIT	O	Extend bus cycle	#
60	RFU		Reserved		2, 3	60	INPACK	O	Input Port Acknowledge	#
61	REG	I	Register select	#	2	61	**REG**	**I**	**Register select & IO Enable**	#
62	BVD2	O	Battery voltage detect 2		2, 3	62	**SPKR**	**O**	**Audio Digital Waveform**	#
63	BVD1	O	Battery voltage detect 1		2, 3	63	**STSCHG**	**O**	**Card Status's Changed**	#
64	D8	I/O	Data bit 8			64	D8	I/O	Data bit 8	
65	D9	I/O	Data bit 9			65	D9	I/O	Data bit 9	
66	D10	I/O	Data bit 10			66	D10	I/O	Data bit 10	
67	CD2	O	Card Detect 2	#	2	67	CD2	O	Card Detect 2	#
68	GND		Ground			68	GND		Ground	

Notes:

Active "low" signals are indicated by a pound (#). Active "high" signals are indicated by a plus (+).

I: Input to card
O: Output from card
I/O: Bi-directional

Glossary

Adapter: The hardware which connects a computer bus to PC Card sockets.

Aliased cards: A memory card that places its Attribute Memory space in the Common Memory.

Attribute Memory: PCMCIA / JEIDA standard PC Cards provide a separate memory address space for recording fundamental card information. This memory is intended to be used by the card manufacturer to record basic configuration information. This memory is selected by asserting the REG# line on the card interface. It is typically, but not necessarily, read-only.
Attribute Memory space need not be physically distinct from common memory space; but it must be logically distinct.

Basic Compatibility Layer: The layer of this standard (layer 1) which mandates the use of a card-information structure (CIS) at the beginning of any complying card.

big-endian byte order: A means of specifying the order in which multi-byte numeric objects are recorded, when broken into bytes. Big-endian byte order specifies that the most-significant byte shall be recorded in the lowest byte address; bytes of decreasing significance shall be recorded sequentially in subsequent bytes. Compare (Cf.) little-endian byte order.

block: For disk-like data formats, a block is the fixed-length sequence of bytes. In such formats, data must usually be read or written as a series of one or more blocks.

buffer page: A region of memory on a card used to improve reliability when updating a card. A buffer page typically includes an indication of the region of the card being updated, an image of the desired value for the region of the card, and a flag that indicates that the buffer page is valid.

byte mapping: The sequence in which byte data is recorded on cards. For 8-bit memory cards, the byte mapping is one-to-one, and not at issue for standardization. For 16-bit and wider cards, the byte mapping within words of the card is arbitrary, and so is governed by the PCMCIA standard.

byte: In the PCMCIA standard, a byte is eight bits.

cache memory: A special fast section of random-access memory (RAM) set aside to store the most frequently accessed information stored in RAM.

Card Information Structure: A data structure written at the beginning of every card that complies with the PCMCIA standard, containing information about the formatting and organization of the data on the card.

Card Services: A software interface specification that allows PCMCIA-aware device drivers, configuration utilities and application programs to share PC cards, sockets and system resources.

checksum: An arithmetic error-checking code for data recording based on summing the bytes of data to be checked. Checksums are frequently used by systems that perform error-checking in software.

CIS:	Card Information Structure.
Client:	A user of Card Services functions. May be a device driver, utility program or application.
Common Memory:	PCMCIA/JEIDA standard cards provide two memory address spaces. The term "Common Memory" denotes the primary address space, containing the memory used for application data storage. See also Attribute Memory.
CRC:	Cyclical Redundancy Check.
cyclical redundancy check:	An error-checking code for data recording based on bitwise polynomial division of the data bytes to be checked. As used in the PCMCIA standard, refers to the 16-bit SDLC version of this code, using the polynomial x16 + x12 + x5 + 1, with the check-register initialized to all ones. CRCs are typically used by systems that perform error-checking in hardware.
cylinder:	A unit of disk organization. A disk is typically viewed as a collection of cylinders. Each cylinder on a disk is divided into tracks; each track is further divided into sectors. Typically, all of the sectors within a cylinder can be accessed without moving the arm of the disk. See sector.
data organization layer:	The layer of the PCMCIA standard covering the data organization of the card.
data organization:	The logical organization of data on a card, independent of the data-recording format. The data organization of a memory card will almost always be some kind of file system.
data-recording format:	The organization of a memory card into sequences of bytes that are updated or accessed by a single logical operation. The data-recording format of a card includes such details as whether the card's data is organized into blocks of bytes; whether the card includes error checking codes for each block; and so forth. The data-recording format does not specify whether a file system is used. The data-recording format of a card is akin to the physical format of a diskette. Cf. data organization.
DOS:	The disk operating system for 80x86 architecture systems, such as the IBM PC. DOS is available in several different versions, which are largely compatible with each other; the term generically designates all of them.
EDC:	Error-detection code.
EEPROM:	Electrically Erasable Programmable Read-Only Memory. A non-volatile memory device which can be programmed electrically, and in which individual bytes can be erased electrically. Usually writes and erasures are much slower than reads.
EPROM:	Erasable Programmable Read-Only Memory. A memory device which can be programmed electrically, and erased in bulk by some means, usually be exposure to ultraviolet light.
error-detection code:	A numeric code derived from the contents of a data block, used to determine whether the data read from the block are probably correct.

file system: An operating-system specified method of structuring data on a mass storage device. A file system standard consists of a set of data structures and the rules by which those structures are interpreted. We sometimes say that a card has a file system recorded on it; by this we mean that an operating system utility program has placed the appropriate information on the card, allowing the card to be interpreted and manipulated by the operating system.
Not all PC Cards have file system on them. Some cards are managed directly by the application program.

Flash EPROM: A type of EPROM that can be electrically erased. It differs from EPROM in that generally the entire memory device must be erased at once (bulk erase). Some Flash EPROM device's memory space is divided into sectors that can be erased individually (sector erase). For example, the Am28F010 is a 1 Mbit bulk erase device whereas the Am29F010 is a 1 Mbit (128 Kbyte) sector erase device that has eight 16 Kbyte sectors.

Flash File System: A file system that can be installed in addition to the DOS File Allocation Table (FAT) file system. The purpose of a Flash File System is to allow an end user to treat a Flash memory card like a floppy disk in his PC.

FlaSh: A trademark of Microsoft, describing a Flash File System designed for use with UV-erasable or Flash EPROM memory cards.

Host: The computer which contains an adapter with one or more sockets.

ISO 646 IRV: International Standards Organization standard number 646 (Character codes), International Reference version. A character set very similar to ASCII, used internationally for representing textual information. It differs from ASCII only in that code 24h represents the international currency symbol rather than the dollar sign ("$"). Except in the alternate/national string tuple, all character data shall be represented using the printing characters from this character set.

Kbyte: kilobyte. 1 Kbyte = 1024 bytes.

little-endian byte order: A means of specifying the order in which multi-byte numeric objects are recorded, when broken into bytes. Little-endian byte order specifies that the least-significant byte shall be recorded in the lowest byte address; bytes of increasing significance shall be recorded sequentially in subsequent bytes. Cf. big-endian byte order.

LSB: Least-significant byte.

Memory Technology Drivers: An embedded or installable device driver whose purpose is to perform the device-specific memory Read/Write/Copy/Erase functions for Card Services.

Metaformat: In the PCMCIA standard, the word metaformat is used to encompass the contents, layout, and interpretation of the card information structure. The PCMCIA Metaformat Standard is outlined in Section 5 of that document.

Micro Channel Architecture: An IBM expansion bus of the type incorporated in some of the personal computers in the PS/2 lines. Features 32-bit addressing and bus-mastering capabilities. Not compatible with ISA or EISA.

one-time programmable: A term describing memory that can be programmed to a specific value once, and thereafter cannot be changed (or can only be revised in a limited way). One-time programmable EPROMs are ordinary EPROMs that have been packaged in such a way that ultra-violet light cannot be used to erase the contents of the EPROM.

OTP: One-Time Programmable.

page: A subdivision of a window. If there is more than one page in a window, all pages are 16 KBytes in size.

paragraph: On 80x86 family machines, a paragraph is a block of sixteen bytes, aligned on a sixteen-byte boundary.

partition: A region of a mass storage device. In the PCMCIA standard, partitions are used to allow a single card to contain two different kinds of data; for example, a card might contain a normal DOS file system in one partition, and directly-executable ROM images in another partition. Most RAM cards will contain only a single partition that contains all the usable storage of the device.

partition check code: A simple method of verifying the contents of an entire partition. A checksum is computed by summing together all the data bytes of the partition; this sum is compared to a value stored in the format tuple that defines the partition. This method is typically used for partitions that change relatively infrequently, such as data partitions in OTP memory.

PCC: Partition check code.

PC Card: A memory or I/O card compliant with the PCMCIA standard.

Protected Mode: In 80286 and later microprocessors, an operating mode in which programs running simultaneously cannot invade each other's memory space or directly access input/out devices, preventing system failures during multitasking operations.

PSP: Program-segment prefix. Under DOS, the PSP is the primary data structure for a process, containing its command line, information about exception handling, and so forth.

Real Mode: An operating mode of the X86 microprocessors in which a program is given a definite storage location in system memory and direct access to peripheral devices. System failure may result if more than one program is loaded in to memory simultaneously.

Reduced Voltage Operation: A PCMCIA standard 2.0 feature that allows the host system to reduce its operating voltage (Vcc) to 3.3 V from 5.0 V after verifying from the PC Card's CIS that the PC Card can operate at 3.3 volt.

Region: A homogenous area of memory on a PC Card using one type of memory devices.

reserved: As used in the PCMCIA standard, a reserved field or code value is set aside for use in future standardization. Vendors shall not use reserved fields or code values for any purpose except compliance with future versions of the PCMCIA standard.

sector: As used in the PCMCIA standard, a sector is the fundamental data storage unit of a disk. A sector is the smallest unit of data that can be individually read or updated. Disk sectors correspond to memory card blocks.

Socket: The 68-pin socket a PC Card is inserted in.

Socket Services: A software interface to the hardware used to manage PC Card sockets in a host computer. It is the lowest layer of the multi-layer software architecture that manages PC Card resources.

TPL: Abbreviation used in symbolic codes to represent the word "tuple".

TSR: Acronym for terminate-and-stay-resident. Under DOS, a TSR is a program that is loaded semi-permanently into memory, extending the system's functionality.

tuple: In the PCMCIA standard, a tuple is a block that appears in the Card Information Structure. Tuples are used to record various items of information about the card layout. All tuples have a common format.

XIP: Acronym for eXecute-In-Place. Refers to specification for directly executing code form a PC Card.

vendor specific: In the PCMCIA standard, this term indicates bits, fields, or code values that are specific to a particular vendor and are not defined by the PCMCIA standard. The PCMCIA standard further distinguishes two kinds of vendor: the card manufacturer, and the supplier of the card data contents.

window: An area in a host computer's memory or I/O port space through which a PC Card may be addressed.

word: As used in the PCMCIA standard, a word is the smallest addressable unit of a given card. Eight-bit cards have eight-bit words composed of one byte; 16-bit cards have 16-bit words, composed of two bytes; and so forth.

Index

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(310) 645-1524
Newport Beach	(714) 752-6262
Sacramento(Roseville)	(916) 786-6700
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 222-7800
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft. Lauderdale	(407) 361-0050
Orlando (Longwood)	(407) 862-9292
GEORGIA	(404) 449-7920
IDAHO	(208) 377-0393
ILLINOIS,	
Chicago (Itasca)	(708) 773-4422
Naperville	(708) 505-9517
MARYLAND	(301) 381-3790
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY,	
Cherry Hill	(609) 662-2900
Parsippany	(201) 299-0002
NEW YORK,	
Brewster	(914) 279-8323
Rochester	(716) 425-8050
NORTH CAROLINA	
Charlotte	(704) 875-3091
Raleigh	(919) 878-8111
OHIO,	
Columbus (Westerville)	(614) 891-6455
Dayton	(513) 439-0268
OREGON	(503) 245-0080
PENNSYLVANIA	(215) 398-8006
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 376-8084

International

BELGIUM, Antwerpen	TEL	(03) 248 43 00
	FAX	(03) 248 46 42
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
GERMANY,		
Bad Homburg	TEL	(06172)-24061
	FAX	(06172)-23195
München	TEL	(089) 45053-0
	FAX	(089) 406490
HONG KONG,	TEL	(852) 865-4525
Wanchai	FAX	(852) 865-4335
ITALY, Milano	TEL	(02) 3390541
	FAX	(02) 38103458
JAPAN,		
Atsugi	TEL	(0462) 29-8460
	FAX	(0462) 29-8458
Kanagawa	TEL	(0462) 47-2911
	FAX	(0462) 47-1729
Tokyo	TEL	(03) 3346-7550
	FAX	(03) 3342-5196
Osaka	TEL	(06) 243-3250
	FAX	(06) 243-3253
KOREA, Seoul	TEL	(82) 2-784-0030
	FAX	(82) 2-784-8014

International *(Continued)*

LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
SINGAPORE	TEL	(65) 3481188
	FAX	(65) 3480161
SWEDEN,		
Stockholm area	TEL	(08) 98 61 80
(Bromma)	FAX	(08) 98 09 06
TAIWAN, Taipei	TEL	(886) 2-7153536
	FAX	(886) 2-7122183
UNITED KINGDOM,		
Manchester area	TEL	(0925) 830380
(Warrington)	FAX	(0925) 830204
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196

North American Representatives

CANADA	
Burnaby, B.C. – DAVETEK MARKETING	(604) 430-3680
Kanata, Ontario – VITEL ELECTRONICS	(613) 592-0060
Mississauga, Ontario – VITEL ELECTRONICS	(416) 564-9720
Lachine, Quebec – VITEL ELECTRONICS	(514) 636-5951
ILLINOIS	
Skokie – INDUSTRIAL REPRESENTATIVES,INC	(708) 967-8430
INDIANA	
Huntington – ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
Indianapolis – ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 469-1312
Wichita – LORENZ SALES	(316) 721-0500
KENTUCKY	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
MICHIGAN	
Holland – COM-TEK SALES, INC	(616) 335-8418
Brighton – COM-TEK SALES, INC	(313) 227-0007
MINNESOTA	
Mel Foster Tech. Sales, Inc.	(612) 941-9790
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICO	
THORSON DESERT STATES	(505) 883-4343
NEW YORK	
East Syracuse – NYCOM, INC	(315) 437-8343
Hauppauge – COMPONENT CONSULTANTS, INC	(516) 273-5050
OHIO	
Centerville – DOLFUSS ROOT & CO	(513) 433-6776
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Westlake – DOLFUSS ROOT & CO	(216) 899-9370
OREGON	
ELECTRA TECHNICAL SALES, INC	(503) 643-5074
PENNSYLVANIA	
RUSSELL F. CLARK CO.,INC.	(412) 242-9500
PUERTO RICO	
COMP REP ASSOC, INC	(809) 746-6550
UTAH	
Front Range Marketing	(801) 288-2500
WASHINGTON	
ELECTRA TECHNICAL SALES	(206) 821-7442
WISCONSIN	
Brookfield – INDUSTRIAL REPRESENTATIVES,INC	(414) 789-9393

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE & LITERATURE ORDERING • TOLL FREE: (800) 222-9323 • (408) 749-5703

17515A 2/3/93
CC-1M-2/93-0 Printed in USA