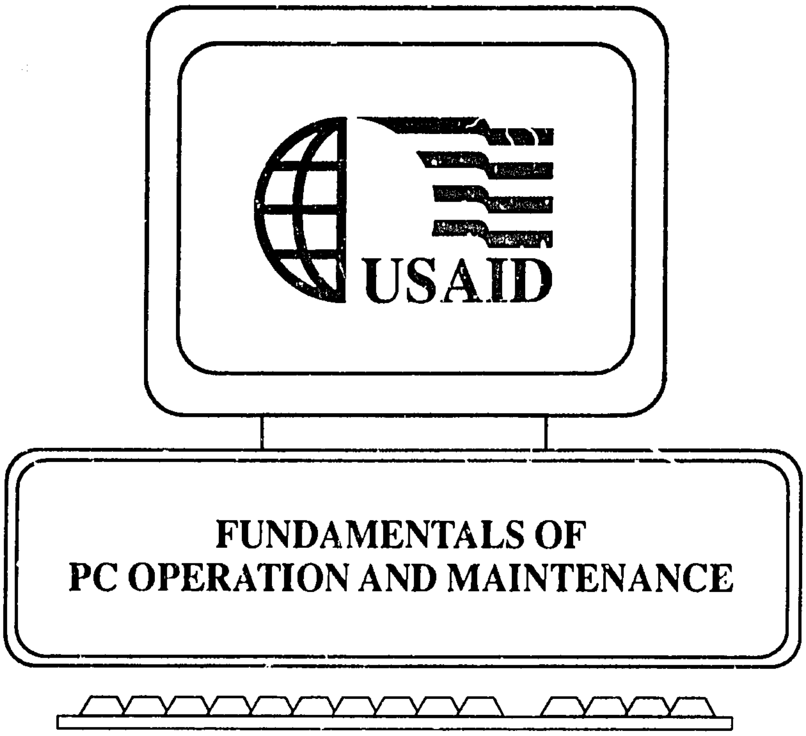


FUNDAMENTALS OF PC OPERATION AND MAINTENANCE

U.S. Agency for International Development
Office of Information Resource Management

January 1992

ACKNOWLEDGMENTS

This manual was partially funded by regional program funds from the ENE, LAC, AFR, APRE and S&T Bureaus of U.S.A.I.D. These Bureaus have provided funds to support AID/FA/IRM Project and Technical Support group information technology-based (IT) activities, such as this manual and the ensuing workshop, for the benefit of all AID program activities. The following PIO/Ts have contributed funding to this effort: #598-0000-3-1654020, #398-0249-3-1632656, #499-0000-3-1672244, #698-0510-3-1611017 and #1361723.

IRM wishes to recognize the aforementioned Bureaus for their funding support of these critical overseas IT transfer and technical support activities. If you have any suggestions or would like to receive additional copies of this manual, please contact FA/IRM/CLS: Joe [illegible]ueron at 703-875-1736.

This manual was produced by Executive Resource Associates, Inc. (ERA), 2011 Crystal Drive, Arlington, Virginia, under contract to the U.S. Agency for International Development.

The manual was written and illustrated by Dr. Raymond. E. Reilly. The author wishes to extend sincere appreciation to Bill Miller and Chris [illegible]ouland for their technical review of the manuscript, and to Amy Barden for her editorial review. The author also thanks Shirley Lindenburg, Tony McWilliams, Ron Nicholson, and Lavonne Adams for their help in producing this manual. Any errors or oversights, however, remain the sole responsibility of the author.

TABLE OF CONTENTS

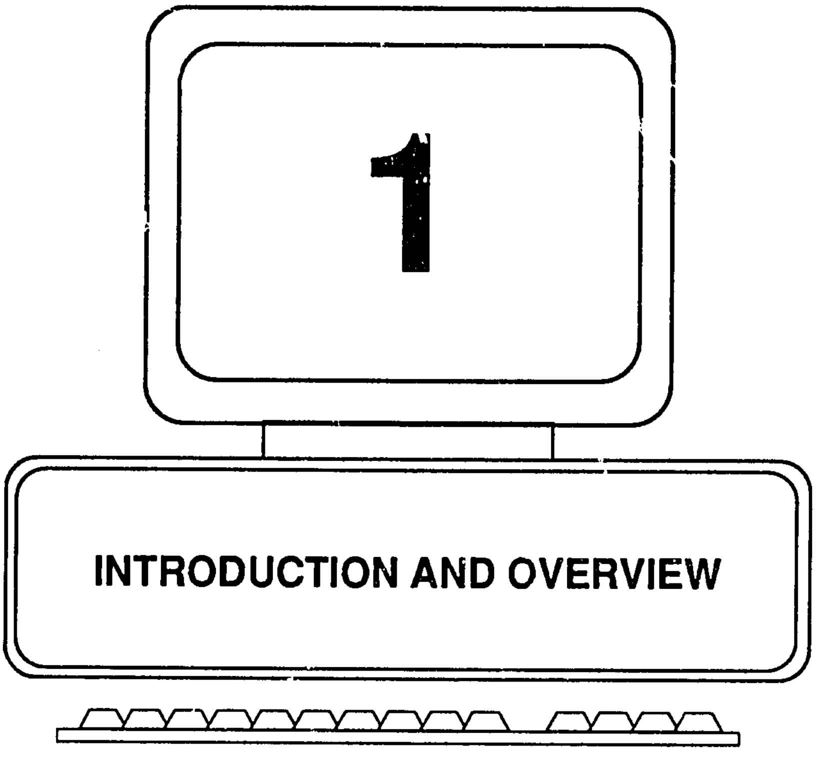
LIST OF FIGURES

Chapter/Figure Number		Page

Chapter/Figure Number **Page**

Chapter/Figure Number		Page

Chapter/Figure Number		Page



1

INTRODUCTION AND OVERVIEW

CHAPTER 1: INTRODUCTION AND OVERVIEW

Introduction

This book was written to help the average personal computer (PC) user and the beginning level support person to understand the fundamentals of PC operation and maintenance. It is intended as a primer for using more technically advanced books and manuals dealing with PCs.

There is no question that at certain levels of troubleshooting and repair, personal computers are extremely complex, requiring considerable expertise and sophisticated equipment to effect a solution. At the same time, PCs are very reliable. Many common problems can be prevented or resolved without having to become a computer expert or an electronics engineer. There are many excellent books available on the subject of PC diagnosis, repair and upgrading, and system configuration. Several are listed at the back of this book.

But for many PC users who are just now becoming comfortable with applications such as word processing or spreadsheets, these reference books may be at first somewhat intimidating. There is so much information, so much unfamiliar terminology, and such a vast sea of details that is it easy to become frustrated.

At a lower level of detail there are books for those who are about to buy their first PC. These publications also serve a very useful purpose, mainly by pointing out major features, options and prices available. But, between these introductory books and the more advanced works mentioned above, there appears to be somewhat of a gap. The books on "how to select a PC" are not primers for diving into a book on "how to repair a PC." For the most part, advanced users of PCs, PC technical support personnel, and even expert authors in the field have achieved that status on their own. Computer Science majors, for example, receive very little exposure to PCs, although their education may, in time, be brought to bear on these "lowly" devices. The average PC user is still left to "read the manual" and struggle along as best he or she can.

In working through just about any book (or an application user's guide provided with a specific product), the reader is unavoidably sent on a tortuous series of digressions: "If you don't understand this or that, consult your DOS manual, or your printer manual, or the documentation that came with your PC" (most likely misplaced by now, if it ever arrived at all). Most of the referrals take you to material that is highly specialized, and which introduces yet another vocabulary and set of concepts to be mastered before you can begin to figure out the problem that brought you there in the first place.

Part of the responsibility for these travails must be assigned to the users themselves. Very often, the need for information arises when a problem occurs in the face of that ever present deadline: "I need it fixed, NOW. I don't have time for theory, or a discourse on the construction of hexadecimal numbers. Just tell how to fix this thing!" And, the answer had better be fast and simple. Over time, a series of these events may occur, but this does not provide an overall conceptual framework for understanding the inner workings of a PC. Neither does it provide a systematic basis for approaching different makes and models of PCs from the standpoint that they are all pretty much alike, varying mainly in their technical details.

The purpose of the present book, therefore, is to provide a generic framework for understanding the functions and components common to all PCs. To the extent allowed by the scope of the book, details are presented for diagnosing and resolving simple problems, for disassembling representative models of PCs, and for effecting basic repairs (mainly by replacing components). With this foundation, it should be much easier to read, understand and apply the information presented in the more detailed reference works.

Overview: Organization and Contents

This book contains 16 chapters, each covering one of the many interrelated aspects of a personal computer. Since a large part of learning about personal computers rests with learning the terminology, a comprehensive glossary is provided at the back of the book. The contents of the remaining chapters are summarized below.

Chapter 2: Basic PC Concepts and Functions

This chapter provides a framework or conceptual overview by which the many parts and pieces of the PC may be more easily related and understood. It recognizes that a PC is a system, much like any other system – an assemblage of components designed to perform a set of specific functions. Every component, chip, printed circuit and peripheral device that make up a PC system falls into one of the three main functions of any system: input, processing, and output.

The hardware, firmware and software are simply the physical embodiments of the desired functions. This is not to say that the system is simple in design or easy to understand. But, by always bearing in mind that each piece of the system has a specific job to do, it is easier to classify the myriad parts and to avoid being intimidated by the often strange and illogical names by which they are identified.

Chapter 3: Primary Components of a PC

Whereas Chapter 2 looks at the kinds of things a PC needs to be able to do, Chapter 3 examines the anatomy of the system. There are many makes and models of PCs in existence, and collectively they present a bewildering array of configurations. At the same time, all PCs share a fundamental anatomy, because all of them essentially perform the same functions: they are tools for creating documents, spreadsheets and databases, and for using other software applications to create a variety of output "products."

They all have a video display, keyboard, disk drive, central processing unit, bus, memory, and power supply. They all can be connected to peripheral devices such as printers and scanners by using interface or adapter boards that plug into slots inside the PC. All of these things are the mechanical and electrical mechanisms for carrying out the machine's intended functions. Knowing in general what these devices are will not make a person into a computer technician, but it will go far toward the ability to recognize the commonalities among machines that otherwise appear to be radically different from one another.

Chapter 4: PC Disassembly

Our progression begins in Chapter 2 with understanding the basic functions of a PC; next, Chapter 3 examines the "brains, arms and legs" – the anatomy of the system. Chapter 4 shows how to perform basic surgery – how to take the machine apart. It covers IBM PCs, XTs, ATs and selected PS/2 Models.

Chapter 5: PC Assembly

Chapter 5 is the reverse of Chapter 4. Rather than asking the reader to put the machine back together by reading the disassembly instructions backwards, Chapter 5 presents all of the assembly steps in the correct order, with the diagrams labeled accordingly. It covers IBM PCs, XTs, ATs and selected PS/2 Models.

Chapter 6: Storing Data on Magnetic Media

The tremendous usefulness of the PC rests much on its ability to store magnetically the results of its (and your) work – the documents, databases and spreadsheets that the users labor so industriously to create. Using and maintaining a PC deals in great measure with understanding how data is written to and read from magnetic media, be it floppy diskette, hard drive or streamer tape. Magnetism is everywhere in a PC. It is in the heads that read and write data; the motors that drive the heads; the surfaces of the disks and tapes that store the data; and the mechanisms that operate an impact printer. It is at the heart of what makes a laser printer print.

H'

So, it behooves any who wish to venture into the inner workings of a PC to have a fundamental understanding of magnetism, and how it is applied in the PC. Chapter 6 provides the groundwork, and perhaps removes some of the mystery from all those bits, bytes, sectors, and cylinders.

Chapter 7: DOS Files and Directories

Chapter 7 explains the basics of DOS, and it also addresses the slippery subject of DOS directories and subdirectories. In order to be able to use applications such as Word Perfect, Lotus 1-2-3, dBase and others, a PC must first have a "master program" for managing its activities, controlling traffic among the various components and devices, and keeping track of where all the applications (programs) and documents are stored. The master program is the machine's "operating system." In the world of IBM PCs, that program is DOS – standing for "Disk Operating System." DOS does far more than its name would suggest – it does not simply "operate disks." It is at the center of everything the PC user does.

DOS has never been very "user friendly." This has kept many users from even attempting to learn about it. Nonetheless, any serious PC user must eventually learn at least the basics of DOS. All the more so, if one is to make the most of the system and to do even routine diagnosis and maintenance.

Chapter 8: Decimal, Binary and Hexadecimal Numbers

Chapter 8 explains the structure and use of hexadecimal numbers and binary numbers by showing their relationship to decimal numbers. Everybody knows how to use decimal numbers. We don't have to go around saying things like. "I am 33 years decimal of age," or "I earn $200 decimal per week." The "decimal" is understood – so much so that everyone takes it for granted and never even mentions it.

The world of computers, however, uses other kinds of numbers, notably binary and hexadecimal numbers. Here the user is faced with strange looking strings of characters such as "100001111," "A000h," and &H10FF. When you finish with Chapter 8, you will know why 100 binary is "really" just the same as plain old decimal "8," and that 3CFh is nothing more than 303 (decimal) in disguise. If you are thinking about skipping the chapter, be aware that memory addresses in a computer are expressed as hexadecimal numbers, and there is much work to be done with the subject of memory.

Chapter 9: Preventive Maintenance

Preventing problems is far preferable to fixing them. Unfortunately, many people find that out only after a disk has crashed and all of that precious data cannot be replaced. Chapter 9 outlines the elements of an effective preventive maintenance program; lists some simple steps that can prevent problems; and it covers the subject of disk backup procedures in some detail.

Chapter 10: Basic Troubleshooting

Chapter 10 offers a basic philosophy or rationale for troubleshooting a system. It then prescribes some fundamental steps for examining a faulty PC. With a little care and patience, many otherwise paralyzing troubles can be cleared up with a minimum of time and effort expended. The subject of identifying the cause of problems is carried forward in the next chapter.

Chapter 11: System Diagnostics

System diagnostics refers generally to the use of tests built into the PC and external software programs to evaluate the operational status of a machine, and to identify specific problems when they exist.

In Chapter 11 we examine the Power-On Self-Test (POST), and review in detail the functions of a comprehensive diagnostics package. The chapter covers system configuration data and CMOS tables; disk drive parameter tables; tests for all of the system's main functions; and benchmark performance tests.

Chapter 12: Installing Adapter Boards

Chapter 12 explains how to set up (configure) an adapter board and install it in a PC. Every peripheral device that connects to a PC (disk drives, video display, keyboard, printer) needs an adapter board (or the equivalent functions stored in ROM). Adapter boards (also called "interface boards," "controllers," or "ports") need to be assigned an IRQ (Interrupt Request Channel) and a Direct Memory Access Channel (DMA).

They use I/O (Input/Output) addresses and they use addresses in RAM (Random Access Memory) for their ROM (Read-Only Memory). Together, Chapters 4, 5 and 12 provide a basic understanding of how "plug-in" options are configured and installed.

Chapter 13: Installing Disk Drives

This chapter covers the steps for drive configuration, physical installation, and system configuration needed to install floppy drives and hard drives. It is easy to physically install a floppy disk drive. But first there is a lot to do. The Drive Selection Jumper, Media Sensor Jumper, and Changeline/Ready Jumper have to be set; and the terminating resistor has to be taken care of. Then the system has to be told (configured) that the new drive has been installed. These topics are addressed in the chapter. Chapter 13 also discusses some of the main types of drive interfaces (ST-506/412; ESDI; IDE; SCSI). If you have ever wondered what a "Scuzzy" (SCSI) interface is, Chapter 13 provides a clue.

Chapter 14: Installing Memory

In the olden days of the IBM PC and PC/XT, Chapter 12 chapter might have been headed "Chips and DIPs." It explains how and where semiconductor memory chips (system RAM) are laid out on the motherboard; how to decode an error message in order to locate a bad chip; how to configure the system when memory is added; and how to handle and install chips and SIMMS (Single-In-Line Memory Modules).

Chapter 15: Power Supplies and Backups

What does the power supply in a personal computer really do? What happens when the power in the office fails? What's the difference between a "standby" backup power supply and an "uninterruptable" power supply (UPS). The answers are found in this chapter.

Chapter 16: Printers

Covering printers in any detail would take a thousand pages. Here, we make a small beginning. Chapter 16 addresses the basic operating principles of impact and laser printers; how to change a toner cartridge; how serial cables and parallel cables are wired; and how to run the self-tests on a printer. It also explains "escape sequences," which are the codes used for controlling the printer to achieve the desired orientation, pitch and other characteristics of printed output.

Recommendations on How to Use This Book

It is recommended that the chapters dealing with fundamentals (rather than the details covered in the other chapters) be read first. Fundamentals are covered in Chapters 2,3,6,7 and 8. By reading these chapters first, you will be well prepared with concepts and terminology when later delving into the technical details of setting DIP switches, reading error codes, interpreting the output of a diagnostics package, and the other technical aspects of PC operation and maintenance.

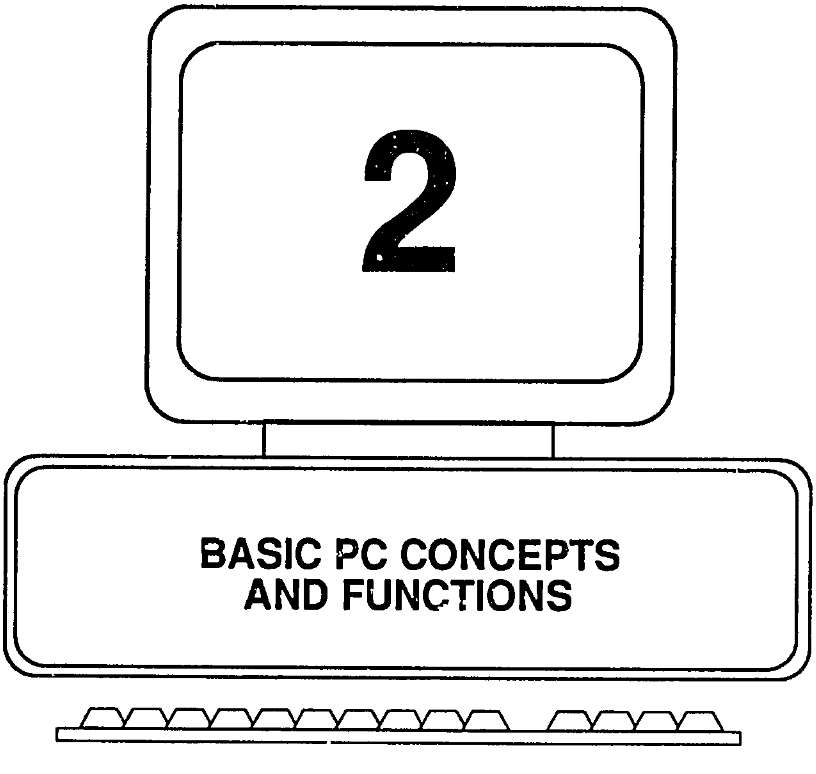
To gain a quick impression of what is in the book, it will be worthwhile to scan the Table of Contents, read the captions in the List of Figures, and review the Glossary presented at the back of the book. Then skim through the book to gain a better idea of what topics are covered, and where they are in the book.

A Word on Conventions

In this book, a personal computer is referred to generically as a "PC." Whenever the IBM PC is mentioned, it is addressed as such. The term "basic" as used in this book means "fundamental," "elementary," or "rudimentary." It has nothing to do with the computer programming language called "BASIC."

The synonyms for various technical and non-technical terms are often provided in parentheses, and the same terms may be defined in several places in the book. This was done to increase the reader's exposure to the unfamiliar terminology, and as an aid to those for whom English is a second language.

NOTES


2
BASIC PC CONCEPTS
AND FUNCTIONS

CHAPTER 2: BASIC PC CONCEPTS AND FUNCTIONS

What Is a Personal Computer?

A personal computer (PC) is an electronic device – a tool for doing calculations; writing programs; and creating text, spreadsheets and graphics, just to name a few of its capabilities. Not many years ago, computers were massive devices, owned and operated by the "data processing department." Your computer work was done for you, as a service provided by computer professionals. In those days you had little control over how and when the computer ran, but neither did you have to be concerned about troubleshooting and maintenance when the computer went down (other than waiting and worrying about when your work would be completed).

Today, the desktop personal computer is commonplace, and its capability surpasses that of many of the giant mainframes of just a decade ago. Your computer is "personal" not only because you have control of it and can use it as you wish, but because you now are more directly and quickly affected by it when it fails to operate properly. The number of PCs in use has greatly outrun the resources available to call upon when a problem arises. Even where help is available, it is certainly frustrating and inefficient to have to wait several hours or more for a support person to arrive, only to have a simple problem solved that, with a little more knowledge, you could have fixed on your own. To acquire that knowledge it may be helpful to understand that a PC is a particular kind of system, and that all systems have certain aspects in common. With a systems frame of reference, it should be easier to see how the various parts of a PC work together to perform the functions of "personal computing."

The PC from a Systems Engineering Viewpoint

A system is a collection of components (which may include human beings) that work together to achieve a specific purpose or perform a particular set of related functions to achieve a common goal. According to this definition, a personal computer (PC) is a system. By understanding the fundamental properties of systems and a few basic principles of electricity and electronics, you will learn more quickly how PCs operate (and malfunction), and you will have a solid framework for recognizing the essential components of any PC, regardless of their form and physical location within the machine.

In designing any system, the first step is to define the functions to be performed. Next is to design (or assemble) and then interconnect the components, and provide the logic that will allow the functions to be executed. All systems, including PCs, have three main operational characteristics: input, processing, and output. The purpose of the system is to achieve the desired output by receiving input and processing it (operating on it or transforming it) according to prescribed rules, and mechanical or electrical events that will produce the results desired. Systems whose output is not fed back as a component of the input, to control the system, are called "open-loop" systems. Systems where the output is sensed and fed back as part of the input are called "closed-loop" systems.

Open-loop Systems

A microwave oven, for example, is an open-loop system. Its input is the food placed inside it, plus the action taken to set the controls (e.g., the cooking time and intensity level) and turn the oven on. Its "processing" function exposes the food to a particular kind of electromagnetic radiation which raises the temperature of the food. Depending on the composition of the food and the time and cooking intensity level selected, the result may be to warm the food, cook it to an edible form, or burn it to a crisp.

Compared to a PC, a microwave oven is a very simple system, mainly because the oven has just one function to perform, which is to generate radiation at the selected level for the time period set on the dial. In fact, most microwave ovens will execute the selected processing function whether or not you place food inside them, or whether or not you place forbidden objects such as a metal dish or aluminum foil in them. In such instances, the oven might easily destroy itself – it has no intelligence, and it does not sense the effects of its actions. In technical terms, it is a open-loop system. It has no "feedback" – its effects or results are not fed back into its input or control elements to adjust them in relation to the output. It just blindly follows its initial instructions (input commands).

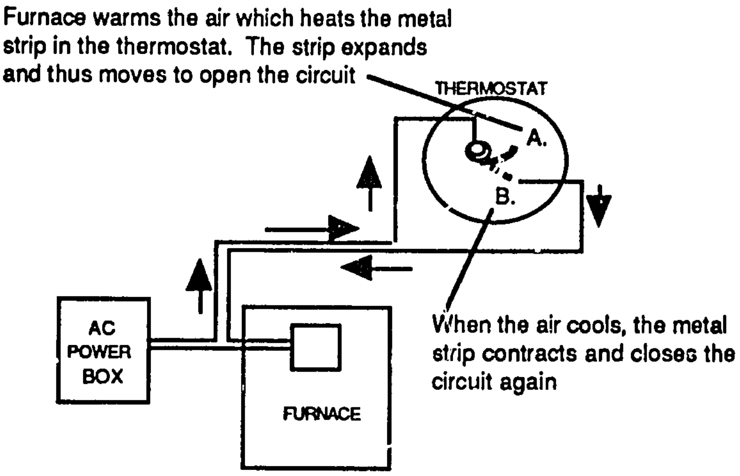
As we shall see later, disk drives operated by a stepping motor work in this way. The motor is told to move the read/write head a certain number of steps toward (or away from) the center of the disk and it does so, blindly. It does not know whether the head is in the exact location desired with respect to the disk, or even whether there is a disk beneath it to be read. There is no communication between the disk surface and the stepping motor. If everything stays in proper mechanical alignment, the device works. If things become misaligned, adjustment by a technician is required.

Closed-loop Systems

A system somewhat "smarter" than the microwave oven is the common thermostat by which you control the temperature in your home (Figure 2-1.) In its simplest form, the device is a switch that turns the furnace on and off. The switch is formed by a piece of metal (actually two strips of different kinds of metal bonded together and rolled up like a spring). When the temperature in the room rises to the selected temperature setting, the metal strip expands and so it moves. In moving, it physically opens the electrical circuit and thus turns off the furnace. As the room gradually cools, the metal strip contracts (shrinks) and thus closes the circuit, once again turning on the furnace.

The heating and cooling cycle continues, and thus the temperature of the room is maintained within an acceptable range. In this system, the output (temperature increase) produced by the furnace, and temperature decrease when the furnace is off) is sensed, and this "information" is fed back to adjust the control mechanism. Such systems are called "closed loop" systems — they have the feature known as "feedback."

Fig. 2-1. The Thermostat: A Simple Closed-loop System



Certain kinds of hard disk drives, called "voice coil" drives, operate on the principle of feedback. One of the read heads and its corresponding disk surface are dedicated to communicating between the disk and the motor that positions the read/write heads that handle the data. Briefly, the dedicated surface has position information on it that is read by the dedicated head. The head passes the information back to the motor, and the motor responds to position the heads according to the information it has received. Thus, in this case, unlike the microwave and the stepping motor drive, there is direct communication between the disk and the motor that positions the heads to read or write information to the disk. This is a closed loop system: it has feedback.

The Human Operator as a System Component

In some systems, notably the PC, the human operator is a major component. The operator generates input (commands) via keyboard, mouse or other device; the computer responds by processing the input (e.g., creating lines of text, doing arithmetic, producing graphics such as drawings or charts). The result (output) is fed back to the operator mainly through a visual display (monitor) or by generating print on paper (hardcopy).

Alternatively, the PC may use an auditory signal generated by an internal speaker to inform the operator that an error has occurred, a command cannot be executed, or that some other noteworthy condition exists. When the computer is started up, for example, combinations of long and short "beeps" are generated to inform the operator when certain error or malfunction conditions exist. The operator senses (hears) and evaluates the output and takes action accordingly. The action may be as simple as correcting a typing error, or as complex as performing extensive diagnostics and replacing one or more components in the system.

As you will see, the notion of open-loop and closed-loop functions is critical to understanding both normal and abnormal PC behavior. Like the thermostat described earlier, the PC constantly verifies, adjusts and calibrates its activity by sensing whether or not certain physical conditions exist and whether certain required information is in place and available.

Sometimes the PC just senses whether a condition, such as an electrical charge, is present. Other times, the PC generates a kind of "test" signal and evaluates the result, for example: was the signal answered quickly enough and in the correct form. Either way, feedback is constantly relied upon. When feedback fails, all manner of bizarre activity may result.

In certain instances, feedback is required from the human operator. One example is in running diagnostics on a color monitor. Here, the PC relies on the human operator to inform it whether the patterns and colors appearing on the screen are as they should be. If the operator says "yes" when the answer should be "no," the PC has no way of sensing the operator's error. It will thus report that the test has been successfully passed.

When a thermostat, microwave automobile or other electromechanical device fails to operate properly, chances are high that hardware or circuitry is at fault. With a PC, however, the first and most likely suspect is the human operator. Next is the software, and last is the hardware or circuitry. Your diagnosis and troubleshooting, therefore, should proceed accordingly.

To become proficient at troubleshooting and remedying PC problems is a long and arduous journey. A certain amount of detail must be committed to memory; a large amount of data must be obtained from appropriate technical manuals and system documentation as the need arises. This assumes that one can even recognize the need, obtain the documentation and finally, read and understand what the documentation says. On the other hand, much time, expense and frustration can be saved by understanding the basic principles of a PC, and being able to recognize the symptoms of simple problems that would otherwise look like disasters. From there, following a slow, careful, methodical and rational approach will usually resolve the problem.

14

Returning now to the idea of a system as a set of components interacting to perform desired "input-processing-output" functions, let's look briefly at the main PC components for each function. Allowing for variations in options installed, all PCs have the same functional components, whether they are built into the PC at the factory, or whether they are attached later as "peripherals" such as monitors, printers, scanners, modems, disk drives or any of the many other devices that may be connected the PC.

Devices for the Functions of Input, Processing and Output

All that varies among makes and models of PCs are the specific details and characteristics of the devices and software employed to carry out each function. We shall address the details later on, but first let's expand our system diagram to include the related major components of a PC. In addition to providing a basic intellectual framework, this will lay a foundation for recognizing components and disassembling a PC, as discussed later.

Figure 2-2 is a simplified list of PC functions at the left, with corresponding basic devices or methods for achieving each, at the right. Figure 2-3 depicts the main components approximately as they may be found in a PC. All components and terminology in the figures will be defined and discussed as we go along. As you will see, it is difficult to classify certain components into one category or another. Some of them serve several functions that cut across input, processing and output. The classification here will thus be somewhat arbitrary, but it will serve our immediate purpose.

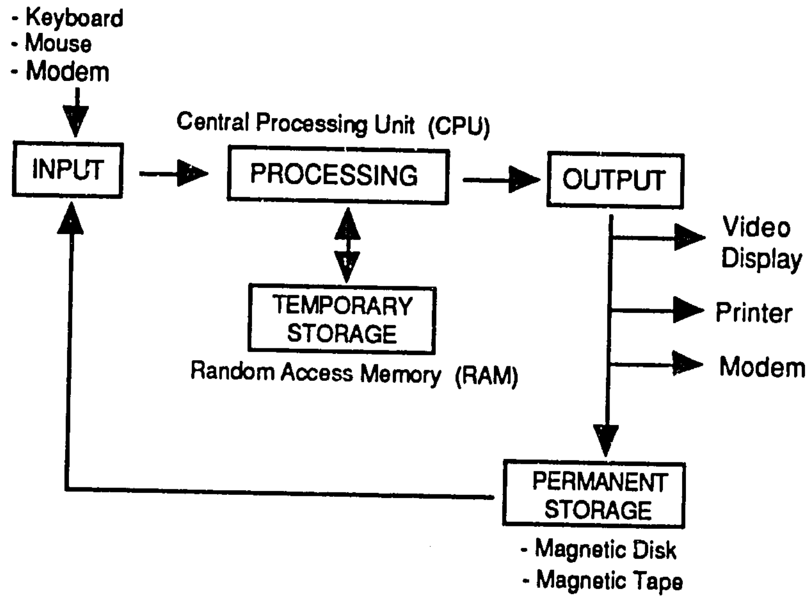
15

Fig. 2-2. PC Functions and Related Devices or Methods

Function	Device/Method
Input – get commands and information into the PC, and activate commands and access information stored permanently inside the PC (e.g., for booting up the system)	Keyboard; mouse; graphic tablet; touch screen; modem; floppy disk; hard disk Read-only Memory (ROM) Basic Input-Output System (BIOS)
Processing – execute commands, operate on data, and control and coordinate all of the inner workings of the machine	Timing crystal/circuit; Central Processing Unit (CPU); Random Access Memory (RAM); Math coprocessor; Date-time clock; Operating system (e.g., DOS); Applications software
Energy supply to operate the system	Power company service; office wiring
Energy conversion and distribution	PC power supply (internal); power conditioners and backups; PC power bus (internal)
Control signal and data distribution	System bus
Output – see and hear what the PC is doing, has done, or is able or unable to do; save and retrieve the output	Display monitor; Lamp and LED displays; Speaker; Disk drives (hard/floppy); Tape drive; Printer; Modem; "Meaningless" noises (troubleshooting)
Expansion – extend or add to the capability of the PC; adjust settings and configuration	Expansion slots and bus; adapter cards and peripherals; DIP switches; jumpers; setup software
Keep track of date and time	Internal clock-calendar
Installation and mounting – hold components in place; make removal/replacement easier.	Chassis; motherboard; circuit boards; brackets; screws, tabs; solder; chip sockets
Environmental control – temperature, humidity, static electricity; prevent dirt and dust	Internal fan (cooling); Control temperature / humidity; Electrical grounding; static electricity discharge paths; keep smoke/ dust away from PC; cover PC; clean/dust PC
Physical Protection – protect the system from physical damage and from intrusion into circuits and components; protect operators and maintainers from physical and electrical hazards	Chassis; case/cover; sealed power supply; electrical ground (power); technician ground strap/wrist; warning labels; safety procedures

16

Fig. 2-3. Input-Processing-Output Functions and Related PC Components



Input

The most familiar devices listed for input are the keyboard and mouse. Input, however, may come from scanners, modems, disk drives, touch screens, graphics tablets, and a host of other devices. Notice that certain devices serve more than one function. For example, disk drives (and their magnetic disks) and modems receive and store output. They are also a source and method of input. Where devices can both provide input and generate output, they are called "input-output" or "I/O" devices.

Information is stored in read-only memory (ROM) chips. A chip is an integrated circuit, packaged in a thin, rectangular casing. ROM chips store information that can be used by the computer, but the PC user cannot add, delete or modify the information. That is why the chip is referred to as "read-only." Although already within the PC, it nonetheless provides input to the CPU when the system is powered on, and it performs other functions during normal operations.

You cannot change the information stored in ROM (without special knowledge and equipment). You cannot use it to store commands or data from your applications software. ROM information is normally changed by replacing the ROM chip itself.

ROM has the advantage of retaining its information when the PC power is turned off. This is in contrast to RAM (Random Access Memory, another form of chip) to be explained shortly) which loses its information when the PC power is turned off.

Also stored in ROM is the PC's Basic Input-Output system (BIOS) which is activated when the machine is turned on, and which starts the sequence of loading special files (programs) from disk so that the PC can operate. Since the startup process involves having one file load, which in turn loads another, which then loads another, the overall process is referred to as "bootstrapping" or simply "booting."

The term bootstrapping is derived from the phrase "to lift one's self by one's own bootstraps." A "cold boot" means to restart the machine by turning the power off and on again. A "warm boot" means to reset the system without turning off the power. This is done by holding down the CTRL, ALT and DEL keys at the same time.

Another ROM chip found in the PC is for the keyboard. It translates keystroke signals (basically, mechanical switch closures on the keyboard) into a form usable by the machine's Central Processing Unit (CPU). The CPU is the "brain" of the computer. The CPU is itself a large, complex integrated circuit (chip).

Finally, since we are concerned with troubleshooting, let's not forget that input also includes electrical power to run the PC, display, disk drives and other peripheral devices. This means also that the associated cables must be connected, and the power switches must be set to the "on" position in order for the input to take place.

Processing

Processing includes all of the internal functions and operations that stand between input on one side of the system and output on the other. It includes not only logical and mathematical operations, but all of the management, coordination, communication and control functions needed to transform input into output.

Central Processing Unit (CPU) and Math Coprocessor

As we have said, the Central Processing Unit (CPU) is the brain of the PC. Although it can perform all of the required logical and mathematical operations, it is often accompanied by a second, optional processor, the "math coprocessor" (also called the Numeric Processing Unit, NPU). Math coprocessors (using a special kind of math called "floating decimal point") speed up calculations involving very large numbers, trigonometric and scientific equations, and processing of numbers associated with displaying graphics. Unless these functions are needed in your work, and unless the software you are running can take advantage of a math coprocessor, the coprocessor will not add to the capability of your machine.

18

Random Access Memory (RAM)

Random Access Memory (RAM), with which we deal at length in Chapters 3 and 14, is the temporary, electronic workspace where commands and data are stored and retrieved during PC use. The name Random Access Memory is misleading, in that it is "memory" only while the PC is powered up. It is a volatile workspace, in that it disappears when the PC is turned off. When the power is turned off, RAM loses all of its information (including any document, spreadsheet or data base that you have not yet saved to disk). Therefore, RAM should not be confused with storage such as that provided by magnetic disk or tape, both of which retain their information in the absence of external power.

Contrary to its name, Random Access Memory is not used in a "random" manner. RAM is provided by chips, each of which, depending on its design, contains thousands or even millions of miniature transistors or capacitors. Information is stored in the chips in the form of electrical charges representing binary numbers. Each chip and the storage areas within it have a unique address (identifier) so that information can be stored and received systematically by referring to the address. This is done mainly by the CPU. But, it is possible for other components to bypass the CPU and write information directly into and retrieve information directly from RAM by using prescribed Direct Memory Access (DMA) channels.

System Timing Crystal and Clock

If the brain of the PC is the CPU, then the heartbeat of the PC is the timing crystal. When supplied with electrical current, the crystal vibrates (oscillates) very rapidly (millions of times per second), and with each vibration it produces an electrical pulse. The rate of vibration depends on the structure of the crystal, and in all cases the rate of vibration is very constant. The crystal thus provides a steady stream of pulses, like the ticking of a clock or metronome. Events within the PC are timed, regulated and coordinated by reference to this "clock."

There is (usually) another clock circuit in the PC, called the "real time" clock. This circuit runs off of a small battery, and it keeps track of the calendar date and time of day. This information is used to mark the date and time that files are created (or last edited). This clock-calendar circuit is independent of the timing ("clock") crystal described above, and the two should not be confused. A PC can run without a real time clock; it cannot operate without a timing crystal.

Interrupt Request Channels (IRQs)

As the CPU goes about its business of performing logical operations and calculations, and transferring information to and from RAM, other events need to take place that require the CPU's attention. Perhaps something needs to be sent to a disk drive or to a printer. For these events to happen, the CPU must be temporarily interrupted from its present task to attend to them. Specific channels are used to request the CPU's attention (interrupt the CPU). They are accordingly termed "Interrupt Request Channels or "IRQs." When peripheral devices such as video or disk drive controller (interface) cards are installed on a PC, it is necessary to set up (configure) the system so that all components know the correct IRQs to use.

Direct Memory Access Channels (DMAs)

As the PC runs, information is normally written to and read from RAM by the CPU, by way of the Basic Input-Output System (BIOS). It is the responsibility of the CPU to manage the use of memory space, to keep track of what is stored where, and to know what space is available for use at any given moment. At certain times, however, it is faster and more efficient for selected components to access RAM without having to run through the CPU and the BIOS. This capability is provided by assigning specific channels (electrical paths) for the information transfer. They are called "Direct Memory Access (DMA)" channels. As with the IRQ channels, the system must be set up (configured) so that the components know which DMA channels to use. Otherwise, conflicts will occur and the PC will freeze up ("hang") or otherwise fail to perform properly.

Operating System

Thus far we have considered some of the basic hardware and electronic circuitry within the PC. But unlike the microwave oven, thermostat and other common electrical appliances, a PC requires much more than hardware and electronic components in order to function. We have seen that input can come from a variety of sources and that output can be sent to a variety of devices. We have seen further that the CPU does the "mental work" of logic and calculations, and that all of this is coordinated by an internal timing circuit (crystal). But in a manner of speaking, all of this is no more than a generic machine or toolbox.

There is essentially nothing within the PC as described thus far that would allow you to run programs and applications, create documents and spreadsheets, or even save anything to a disk. At this point, the PC is much like a symphony orchestra sitting in place, but lacking sheets of music to play from and a conductor to guide its performance. To carry the analogy further, the musicians are capable of reading and playing many different kinds of music, and they can take direction from any one of a number of different conductors. Note, however, that they can respond to only one conductor at a time.

The orchestra conductor, in the form of a collection of special software programs called an "Operating System (OS)," brings the PC orchestra musicians to life. The operating system is thus a software program software that, once loaded itself, allows other programs (applications) to be loaded and used. You then provide the sheets of music in the form of applications such as word processing, spreadsheet and data base programs, and by entering commands via the keyboard or mouse. As these applications are used, output is created in the form of text, spreadsheets, data bases and graphics. The operating system keeps track of these products (files), and it allows you to create directories (like file folders) in which to organize your documents. It also does other tasks associated with creating and managing files, such as naming, copying, and erasing them.

The operating system is responsible also for interacting with and using the many components we discussed earlier. For example, an important function of the operating system is to map ("format") and manage the storage space provided by the disk drives, and to handle requests for writing information to the disk and retrieving information from it. Of course, the operating system relies here on other components, notably the disk controller (interface) card, to access the disks, but the operating system is what orchestrates the activity.

Disk Operating System (DOS)

There are a variety of operating systems available for use, and one or more can be used (one at a time) on a given PC. However, the IBM PC/XT/AT family of machines and their clones are designed primarily to run a particular operating system called "DOS." DOS stands for "Disk Operating System." It derived its name long ago, when disk drives were first introduced to the PC world, replacing the cassette storage scheme seen on early PCs. Although DOS still handles disk operations, it has evolved over the years into much greater capability (and size). Therefore, the name "Disk Operating System" today fails to convey the true scope and power of DOS.

DOS was developed by the Microsoft Corporation, which licenses the basic product, for modification and use, to other companies. When IBM adopted DOS as its standard, DOS became the de facto standard for the PC industry. You should understand, however, that there is no generic or "plain vanilla" form of DOS. IBM uses a form called PC-DOS. IBM clones use MS-DOS (standing for Microsoft-DOS). PCs manufactured by different companies may have different BIOSs, and even different versions of their basic BIOSs, in existence. The form of DOS used must agree with the BIOS used.

Further, within a given form of DOS, such as PC-DOS, there is an evolutionary chain of versions, all the way from PC-DOS 1.0 through, at this writing, PC-DOS 5.0. The operating system is continually being changed in order to eliminate "bugs" (errors), increase capability, or both. In diagnosing illnesses in PCs it will be important for you to know which type and version of DOS your "patient" is running.

Software Applications

We have already mentioned applications software. This is the familiar "package" you buy and install on your PC to do word processing, spreadsheets, data bases, graphics and other productive work. Note that each of these applications packages will specify the version of DOS needed to run it, as well as the minimum amount of RAM required. The version of DOS specified indicates that a given version number, e.g., 3.0 or later (a higher version number), is needed to run the application. Failing either of these, the application either will not run, or it may run but not perform properly.

Output

Just about every electronic and electromechanical component receives some form of input and produces some form of output. For our present discussion, we shall be concerned with output that reaches or passes through the physical boundaries of the PC, and that has use or meaning for the human operator or support person. Thus, information displayed on a monitor, anything written to a disk drive or sent over a modem, and anything printed out on paper is "output." Other forms of output include tones or beeps from the PC speaker and light-emitting diodes (LEDs) or lamps illuminated on disk drive panels and the keyboard.

Besides the formal, deliberate outputs just mentioned, the astute technician will pay heed to the sounds made by a PC during normal operations. (These sounds may generally go unnoticed by the PC user.) For example, can you hear the fan motor running? How do the disk drives sound when they spin? Does the whirring and chattering sound normal, or has it changed? Is there a vibration or a squeak that might indicate trouble on the way? Is the hard drive creating a grinding noise, telling you that it is eating your precious data? Is the drive emitting a squeal that sounds serious, but is really no more than a need to lubricate (or place a tiny dab of silicone putty on) the static discharge wiper on the drive spindle?

The temperature of components is also a factor. Do certain chips feel warmer or cooler than they should be? All of this is a form of output, in the sense that it has meaning to the PC user or maintainer.

Supporting Components (Power and Buses)

Thus far we have looked some of the "high level" components and software that serve the general input-processing-output functions of a PC. Among these, the components most prone to failure are the disk drives. Accordingly we devote considerable attention to disk drives in Chapters 6 and 13.

22

To complete our initial tour of the PC, however, we will look next at two critical, underlying elements of the PC — the power supply, and the electrical pathways ("buses") along which the power and the information flow into, within, and out of the PC. Most of the latter pathways are provided not by wire, but rather by thin, narrow traces of metal etched ("printed") onto a board, thus the term "printed circuits."

The PC Power Supply

Except for laptop computers that run off of batteries, PCs need a source of conventional electrical power in order to operate. Normally this is provided by the wiring within the home or office where the PC is used. We can refer to such electrical service as the "source" of electrical power. This will help to distinguish it from the function of the "power supply" within the PC. The PC power supply does not provide power in the sense of creating it.

The power supply simply converts the power from the wall outlet to a form usable by the PC (i.e., from high voltage alternating current to low voltage direct current). Figure 2-4 shows the basic functions and parts of a power supply. The incoming (source) high voltage AC power is reduced to low voltage power by means of a step-down transformer. The output of the transformer is run through a rectifier, whose function is to convert the flow from alternating its direction to one direction only (AC to DC). Capacitors, which act like electrical storage tanks and filters, smooth the flow of current, eliminating the variations in flow associated with the original AC current.

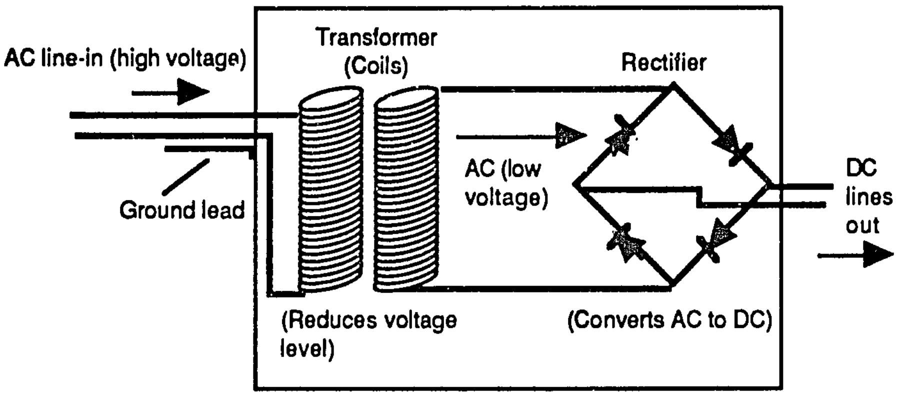
Next to the disk drives, the power supply is the component most likely to malfunction or fail. Unfortunately, when the power supply is not working properly, the PC is useless. Fortunately, however, it is easy to diagnose and replace a faulty power supply.

Electrical power in homes and office buildings is provided in the form of alternating current (AC). As its name suggests, alternating current reverses its direction of flow (it "cycles") many times each second. Depending where you are in the world, the rate of flow reversal (cycling) will normally be either 60 cycles per second or 50 cycles per second. Furthermore, the AC power is delivered at a relatively high level of pressure (voltage), in order to be able to operate appliances that require (draw) a lot of electricity. The most common combinations of AC voltage and cycling rate are 120 volts, 60 cycles per second, and 230 volts, 50 cycles per second.

Cycles per second is usually indicated by the term "Hertz" and abbreviated as Hz. Volts is abbreviated as "v." Thus the two preceding pairs of values will usually be written as 120v - 60Hz; and 230v - 50Hz, respectively. Neither of these, however, is what a PC needs in order to operate its internal components.

23

Fig. 2-4. Simplified Illustration of a Power Supply



As explained earlier, a PC runs on direct current (DC), and uses both 5 volts DC (5vdc) and 12 volts DC (12vdc). Direct current flows in only one direction, so the term "cycles per second" or Hz does not apply. The function of the PC power supply is to convert the 120v - 60Hz, or 230v - 50Hz electrical power into 5vdc (5 volts, direct current) and 12vdc. The 12vdc power is used to run the PC cooling fan motor and the disk drive motors. The 5vdc power is used by the electronic components to provide internal control signals and data processing.

Figure 2-2 referenced power conditioning and power backup devices. These are further discussed in Chapter 15. In brief, these devices may serve one or more of several functions, including "conditioning" the AC power from the wall outlet (filtering out spikes and surges) and/or providing a battery-based supply of power that will allow you to operate the computer for a short time in the event of an AC source power failure. These devices are installed between the wall outlet and the PC power supply. That is, the computer plugs into the device and the device plugs into the wall outlet.

The PC does not operate directly off of the batteries of a backup supply. The battery-based, backup devices convert their DC power to AC and send it to the input side of the PC power supply. The PC power supply converts the AC power to 12vdc (for fan and disk drive operation) and 5vdc current (for logic and data functions). The DC output is connected to controller cards and to the PC motherboard (discussed later). The motherboard then distributes the power (makes it available to other components) by means of the printed circuit paths ("power bus") mentioned earlier.

24

The CPU and other electronic components use the 5vdc current to form electrical charges and pulses to represent and transfer information, and to generate and transmit control signals to the various input and output devices.

Buses and PC Architectures

Where two or more components can share such paths, the paths are referred to as a "bus." The term originated as "omnibus connector," suggesting that various components could attach to or tap into it. Today the term has been shortened to "bus." The AC wiring in your home is a type of bus. The wall outlets (like the expansion slots in the PC) allow you to plug various appliances (lamps, toaster, microwave, television set, etc.) into the bus.

The design and layout of the buses as mentioned above (along with the CPU) is what distinguishes among various genres of PCs. The bus design largely determines the specifications of components that can be attached to or plugged into the PC main circuitry. Thus, the bus design has become synonymous with the term "architecture." Or, perhaps this occurred in the reverse direction, but no matter.

ISA, MCA and EISA Architectures

In brief, the early PCs used a particular bus design. When IBM entered the PC market and adopted that design, the design acquired the name "Industry Standard Architecture" or "ISA." More recently, IBM has come up with a new and different design called "Micro Channel Architecture," or "MCA." MCA is the architecture used in certain models of the IBM PS/2 family of computers. Since this design is proprietary to IBM and thus poses a threat to the competition, certain other companies in the market got together and modified the old ISA to become the "Extended Industry Standard Architecture," or "EISA." Which design will eventually become the "true" industry standard remains to be seen.

PC Options and Adapter Cards

Many persons new to PCs are dismayed to discover that the price quoted in the newspaper ad failed to include certain "options," such as a monitor (plus the needed adapter/controller card, at extra cost), a decent keyboard, and a hard disk drive (plus another adapter/controller card, at extra cost). Of course, the price of that terrific printer failed to include the additional $30 - $80 cost for a cable to connect it to the PC, and a toner cartridge for another $100.

Our real concern, however, is not so much with the hidden costs of purchasing a PC. It is simply that, unlike most household or office appliances that are ready to plug in and turn on, the PC is designed to accommodate a wide variety of add-on features and functions. This is accomplished by "expansion slots" into which the "adapter card" of the "peripheral device" is plugged. With minor exceptions, any adapter can be plugged into any slot, just as different appliances can be plugged into any of the wall outlets in your home.

Since a card (e.g., disk drive controller, video card, graphics adapter) plugs in electrically not only by a strip of contacts along its bottom (its "edge connector") but by one or more cables, one consideration in selecting a slot is in how best to arrange the cables so they are not stretched or cramped. Care should be taken also to see that the cables do not protrude, where they may be snagged and damaged when removing the PC cover.

It also happens that in certain PCs, there are very slight variations in the timing signals (recall the timing circuit discussed earlier) arriving at the slots. Although the signals are supposed to be perfectly synchronized, these infinitesimal differences can result in improper performance. For this reason, a problem can sometimes be cured just by rearranging the adapter cards into different slot positions.

What is an adapter card and why do we need it? A PC is a veritable United Nations of components, and each peripheral device has its own "language." The processing of information by the CPU and RAM circuits is done with very small amounts of power, and the operations rely on binary coded numbers expressed as tiny electrical charges. When the information needs to be recorded on a disk or displayed on a monitor, a "translation" is required.

The adapter card performs the necessary translation essentially by converting signals and data from one electrical form and level to that needed by the peripheral device. Adapters are also called "interfaces" because they stand between two different components or devices and allow the devices to communicate (interface).

System Configuration – Jumpers and DIP Switches

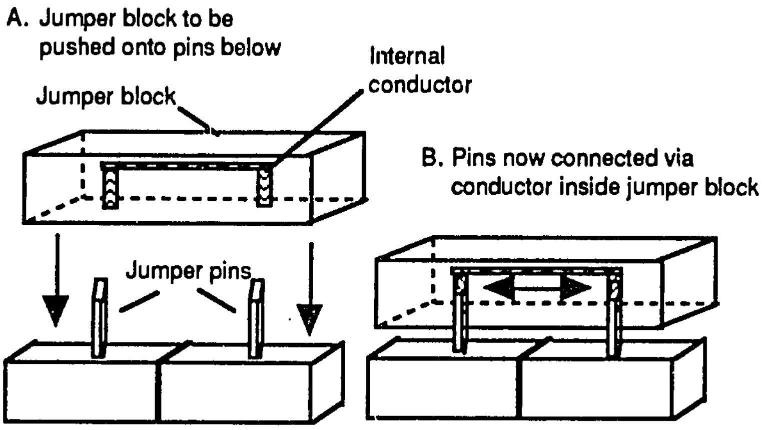
Because of the versatility provided by the design of PCs, with their adapter slots and the host of components that can be plugged into them, it is necessary to "tell" the PC about the device(s) that have been installed. Otherwise, it will not recognize them and, therefore, it will not use them. There are three ways in which the PC can be informed about the devices installed – jumpers, DIP switches, and setup software. Each is described briefly below.

26

Jumpers

A jumper is simply a short electrical conductor (like a piece of metal or a wire), enclosed in a plastic case (Figure 2-5). Holes in the case allow the jumper to mate with pins projecting out from an electrical circuit. For example, given four pins sticking up (pins 1,2,3,4), the jumper may connect pin 1 to pin 3, or pin 2 to pin 4, and so on. By placing the jumper across specific pins, a particular electrical path is thus completed (selected).

Fig. 2-5. Illustration of a Jumper Type Connector

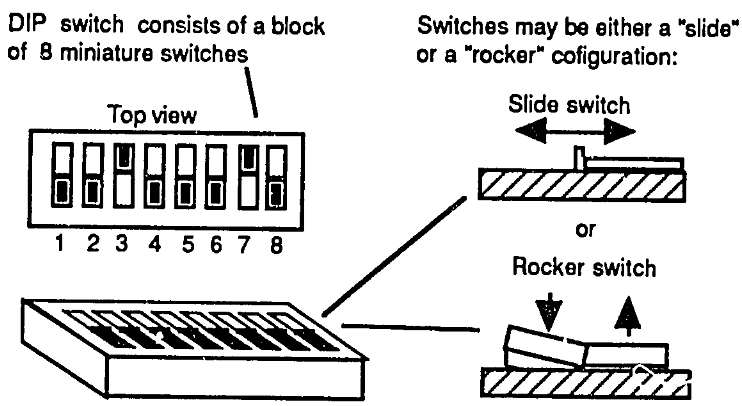


DIP Switches

When there are only one or two "selections" to be made, jumpers are normally used. Where more settings (choices) are required, the wiring is configured by setting the positions on a multiple choice switch, the "DIP" switch. DIP stands for Dual In-line Pin, which reflects the physical design of the switch, which contains pairs of pins aligned to form two parallel rows. Thus, instead of having to use a collection of individual jumpers, you simply set each switch on the DIP to its open or closed position. Typical DIP switch designs are depicted in Figure 2-6.

Certain DIP switches have a small, slide mechanism for this, while other use a "rocker arm" design. They all serve the same purpose. The positions to be set are stated in the documentation that comes with the device. Often, the switches are set at the factory ("default" settings) and changing them is not necessary. But, you should always check that the settings are correct for your system. And, you should make a record (diagram) of the settings before removing or working around these components. If just one DIP switch setting is accidently changed, you could spend hours looking for a "problem" that doesn't exist.

Fig. 2-6. Typical DIP Switch Designs



Configuration Software and CMOS Chips

Whereas the earlier PCs relied much on jumpers and DIP switches, later models (especially the IBM PS/2) family has made all of this unnecessary. In place of mechanical jumpers and switches, software is used to inform the PC about the devices installed. Setup software is provided that displays menus and prompts you to enter the necessary "configuration" information. The information is then stored in relatively permanent form within the machine, so that it need not be entered each time the system is booted up.

The information is stored in a special type of chip called CMOS. CMOS stands for "Complementary Metal Oxide Semiconductor." The chip is provided electrical energy by a small battery, and thus is able to store the information when the PC main power is turned off. You should, however, keep a printed record of the information. If the CMOS battery fails or is disconnected, or if the information is corrupted (it happens), it pays to have a printout handy. Otherwise it may be quite difficult and time consuming to find out all of the values that must be re-entered into the configuration table.

Often, PCs arrive at an office already configured, but no written data is provided. What happens then when a problem arises? To avoid unnecessary work, go to each machine and run ether the setup program (but don't change the data), or run a third party diagnostic such as Checkit. (See Chapter 11 for details.) With the configuration data displayed on the screen, print a copy (use Print-Screen key). If you are using diagnostic software such as Checkit, you have the option of printing the data or saving it to a disk. Then, when the user reports that the PC "no longer recognizes" a device (such as that all-important hard drive), you may be able to solve the problem with just a few keystrokes.

PC Component Mountings and System Enclosures

Everything we have discussed so far, and many other parts and components to be covered later, must be arranged, mounted, connected and held in place. As we shall now see, all of this is done in a way that makes any PC/XT/AT, and PS/2 Models 30 and 50 (desktops) easy to take apart and reassemble. The other PS/2 models are different, but even simpler to take apart and reassemble. PC disassembly and reassembly is covered in Chapters 4 and 5.

Covers

There are exceptions to any rule, but the following conditions apply generally to all desktop PCs. Note that what follows is not a set of disassembly instructions. It is a general description, to provide an overview of how the PC is put together, and to illustrate that stripping down a PC is not a complex task. Detailed disassembly instructions are provided in Chapter 4.

Depending on the model of PC in question, the cover is essentially a U-shaped shell or a set of panels that connect to the frame or chassis of the unit. No components are connected to the cover itself. By removing a few screws at the back (along the edge of the panel) and occasionally at each side, the cover can be removed by tilting it up and sliding it forward. Be careful when doing so not to snag any cables inside. Of course, once the cover is removed the cables to any component should be disconnected before attempting to remove the component itself.

Chassis

The foundation of the unit is the chassis— the metal base, side and rear panel unit over which the cover slides to enclose the entire system. With the cover removed, you will easily see where the power supply and cooling fan are held in place by screws entering through the back of the rear panel.

Power Supply Mounting

Never attempt to take a power supply apart, and never run a power supply when it has been removed from the PC. Power supplies retain a strong electrical charge, even when they are unplugged. Shock from this stored energy is harmful and may even be fatal. Running an unconnected power supply may cause it to explode!

In addition to screws, certain components such as the power supply often will have little "catches" or tabs underneath them. If so, after the screws have been removed, the component must be shifted (slid), usually toward the front of the PC or sideways, and then lifted out. It is never necessary to force a component. If does not come out easily, something is being overlooked.

Disk Drive Mounting

The hard disk drive may be held in place partly by a screw entering from the bottom of the chassis, and further by a bracket assembly within which the drive sits. The same idea applies to the floppy drive. As we shall see later, in the chapter on PC disassembly, the PS/2 models are much different in construction than the PC, XT and AT. The PS/2s are essentially "snap-together" designs that require almost no tools for disassembly.

System Board (Motherboard) Mounting

The large printed circuit board containing the expansion slots; CPU, RAM and ROM chips; timing crystal and other electronic components is the motherboard. IBM's term for this is the "system board." In this book, the terms are used synonymously. The board (PC/XT/AT) sits on plastic "spacers" or "standoffs" to keep it out of contact with the metal chassis panel below it. Adapter cards plugged into the expansion slots on the motherboard are each held in place by a retaining screw at the top/rear of each card. The cards are removed before removing the motherboard. The motherboard itself is held in place by several screws. In removing the motherboard, the spacers must be loosened but should not be removed. The motherboard is usually the last item to be removed when disassembling a PC.

Chip Mounting Methods

On the surface of the motherboard are mounted the CPU, math co-processor (if any), RAM and ROM chips. The RAM chips are easy to recognize. They are arranged in a rectangular pattern of rows and columns. Specific rows are designated as "banks" and banks are numbered in sequence, beginning with 0 (zero).

30

A chip may be plugged into a socket that is soldered to the board, or the chip may be soldered directly to the motherboard. Socketed chips may be removed by prying them up with a small, thin screwdriver, or by using a chip extraction tool. Experts vary on which tool (screwdriver vs. extractor) to use. Some assert that, in the hands of a novice, the extraction tool is more likely to damage the chip.

Chips soldered to the board cannot be removed without special skill. Chips are very sensitive to heat, and the heat from a soldering iron can easily be fatal to them. Generally, where a PC has the first bank of RAM chips (bank 0) soldered to the motherboard, a single bad chip in that bank means that the entire motherboard must be replaced. The operating system (DOS) loads itself into bank 0, and is unable to load itself into other RAM locations (chips) anywhere else. Therefore, if Bank 0 is not working, the PC will not run.

As we have seen, an entire PC can be taken apart with just a few screwdrivers. But it is said that if a fool takes it apart, it will take a genius to put it back together. The problem arises here because the "fool" fails to label the components, cables and connectors, fails to diagram the location and orientation of parts before removing them, and either loses some of the screws or gets them mixed up. Trusting to memory is very risky indeed! It is the genius who labels and diagrams during disassembly.

As a technical support person or as an informed user, taking the PC apart should be done only as a last resort (or as an educational exercise). Better than 90 percent of all problems can be accounted for by user errors, configuration errors, and software errors. These problems should be ruled out before attacking the innards of the machine. Other problems, which arise out of careless use, or neglect of preventive maintenance, can be greatly reduced by following just a few simple procedures as summarized below.

Precautions Concerning Use and Maintenance of PCs

PCs are vulnerable to heat, and in particular, <u>changes</u> in temperature. The latter changes result in repeated expansion and contraction of components, known as "thermal stress." Under such temperature variations, chips can "creep" out of their sockets, internal elements of components can fail, and condensation can form to produce short circuits and corrosion. When electrical components are cold (relative to their normal operating temperature), their electrical resistance is greatly reduced. This allows a harmful surge of power to flow through them when they are turned on. That is when they often will fail.

31

You have probably noticed that most light bulbs fail not when they are burning, but when they are first turned on. The power surge through the cold filament is the reason for this. The same principle applies to a PC. The best form of prevention here is to leave the PC running 24 hours a day, seven days a week. This will also help prevent problems with disk drives (for reasons discussed elsewhere). Also important is to ensure that the internal cooling fan provides adequate air flow. If the fan fails to operate, shut the PC down immediately. Also, ensure that there is no coating of dust on the internal components. Dust is an insulator, and thus contributes to components overheating.

PCs are vulnerable to static electricity, especially when you are handling sensitive components like chips. Static charges build up through friction between moving objects, such as when you walk across a carpet. When you then touch a metal object such as a PC, the charge is released like a miniature lightning bolt.

When humidity is low, static buildup is stronger. Also, dust (and smoke particles) can hold a static charge. So, low humidity, friction due to normal movement, and dust all contribute to potential problems caused by static electricity.

To reduce the risk of damage to the PC by static discharge:

- Discharge yourself by touching a metal doorknob or piece of metal furniture before approaching the PC.
- Remove your shoes (if permissible) when working on the inside of a PC.
- Wear a professional ground strap when working on the inside of a PC; do not attempt to make your own ground strap -- you could receive a serious and even fatal electric shock.
- Keep the PC interior free of dust and dirt.
- Keep the ambient humidity within the range prescribed by the PC documentation; a relative humidity of 50% is a good reference point.

Chapter Summary

In this chapter we have viewed the PC as a system designed to perform specific input, processing and output functions, and we have looked briefly at the hardware and software typically used to carry out those functions. We have observed that the line of demarcation between functions is not always sharp, and that certain hardware is designed to perform multiple functions. Nonetheless, at any given

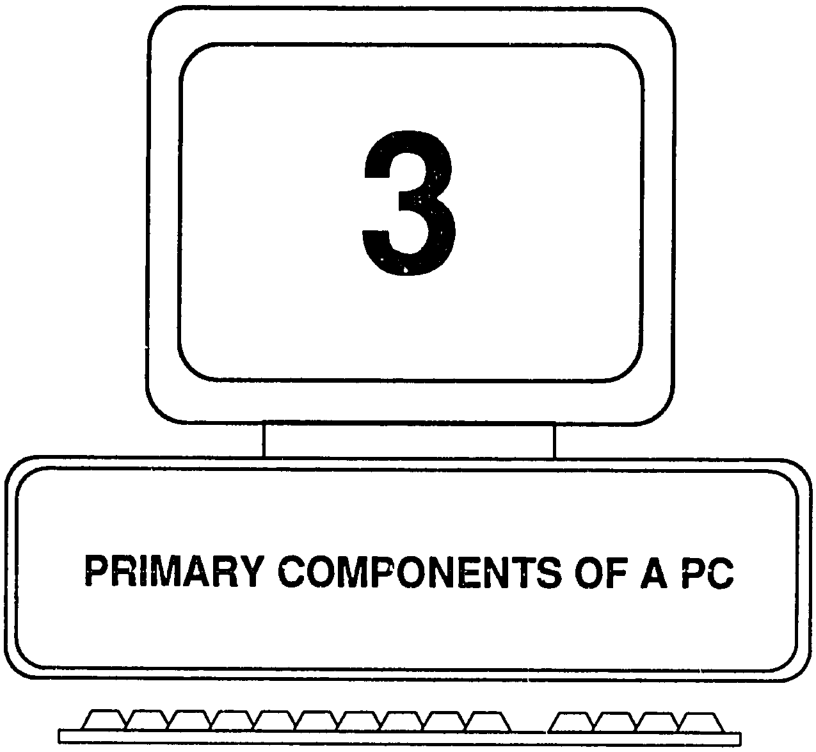
72

moment, a particular device performs only one function, for example, either input or output. Because a computer can operate at a very high speed, functions may be alternated so quickly that they may appear to be happening simultaneously, however.

In summary, a PC represents a clearly defined set of functions or operations, and for each function or operation there is a piece of hardware of software to carry out that task. The components are mounted on boards and/or attached to a metal support structure (chassis); and the unit is enclosed by a cover, to which components usually are not attached.

Most important to gain from all of this discussion is the idea that all PCs perform the same basic input-processing-output functions, and all PCs therefore have corresponding hardware and software as the physical embodiment of those functions. When you first open up a PC, the scene may appear rather complex and confusing because there appear to be so many components, cables and connections all crammed into a relatively small space. But if you keep your "systems perspective," you will quickly learn to identify the physical components in terms of the tasks they perform. In the next chapter we look at the components and their layout in more detail.

NOTES



3

PRIMARY COMPONENTS OF A PC

CHAPTER 3: PRIMARY COMPONENTS OF A PC

In the preceding chapter we approached the PC from a systems perspective. We noted that a PC is a machine designed to generate a particular type of output in response to input from the user, and that the input is converted to output by means of a collection of functions termed "processing." All computers are similar in that they are designed and operate according to this elementary input-processing-output model.

The many functions to be accomplished, of course, must be embodied in physical form, i.e., mechanical, electrical and electronic components, along with an arrangement of electrical pathways for providing operating power and transmitting control signals and data. Our next step is to explore the primary physical parts of a PC, and to see in general how the system is laid out. From this chapter and the one preceding, you should be able to list from memory the basic functions and components of any PC, and you should be able to identify most of these components when you open up a machine, as we shall do in the next chapter.

The PC as a Modular Device

The PC is a modular device. That is, it is an assembly of relatively self-contained units; and, it is designed to accept a variety of options that are attached by connectors and cables. There is great variation within and across PC makes and models. This complicates the process of upgrading, troubleshooting and repair. In all cases, it is necessary to have the documentation (whether printed or on disk) of the specific PC to be worked on, and the documentation provided with each option to be installed on the PC.

Each optional device or component has its own particular specifications and operating characteristics. Replacing or upgrading an option (anything from a display or hard disk drive down to a single chip) requires knowledge of the type or class of device in relation to the PC in which it may be installed, and on many occasions, a detailed understanding (interpretation or "decoding") of a specific part number.

A detailed listing of components and parts is beyond the scope of this introductory text. A wealth of such information, however, may be found in the references listed at the back of this book. This chapter provides an overview of the main PC components, the functions they perform and their general location within several representative systems. With this information, you should be able to inspect even an unfamiliar PC and identify its main components. Then by referring to the system documentation, information printed on the PC motherboard, and/or on the component, you should be able to identify the component precisely. At that point you will be able to replace the component with an identical part (either new or obtained from a defunct PC). Given the specifications of the PC make and model in question, you will further be able to identify compatible replacements or upgrades by reference to technical publications or equipment vendors.

The Main Components and Layout of a PC

The main components of a PC are listed below. Each one is then discussed briefly in turn.

- Motherboard (System Board) contains:
    - System Clock (Timing Circuit)
    - CPU (Central Processing Unit)
    - Math Coprocessor (Numeric Coprocessor)
    - Bus
    - Expansion Slots
    - Main Memory (Planar Memory)
    - Keyboard Adapter

- Power Supply
- Keyboard
- Display (Monitor) and Display Adapter
- Floppy Disk Drive(s) and Controller
- Hard (Fixed) Drive(s) and Drive Controller
- Multi-function Board
    - Printer Port
    - Clock Calendar
    - Serial Port (RS 232C Port)

Motherboard (System Board)

The motherboard (system board) is the main circuit board of the PC. It holds the components as described briefly, below. Figures 3-1 through 3-4 depict the basic layout of the motherboard for the PC, XT, AT, and a typical 80386 machine, respectively. Note the many common components across the different boards, even though the boards are laid out differently.

System Clock (Timing/Oscillator) Crystal

The clock crystal provides the "heartbeat" or "metronome" by which the system, in particular the CPU, operates and coordinates its functions. The crystal operates in conjunction with the system clock chip.

Fig. 3-1. Basic PC System Board Layout

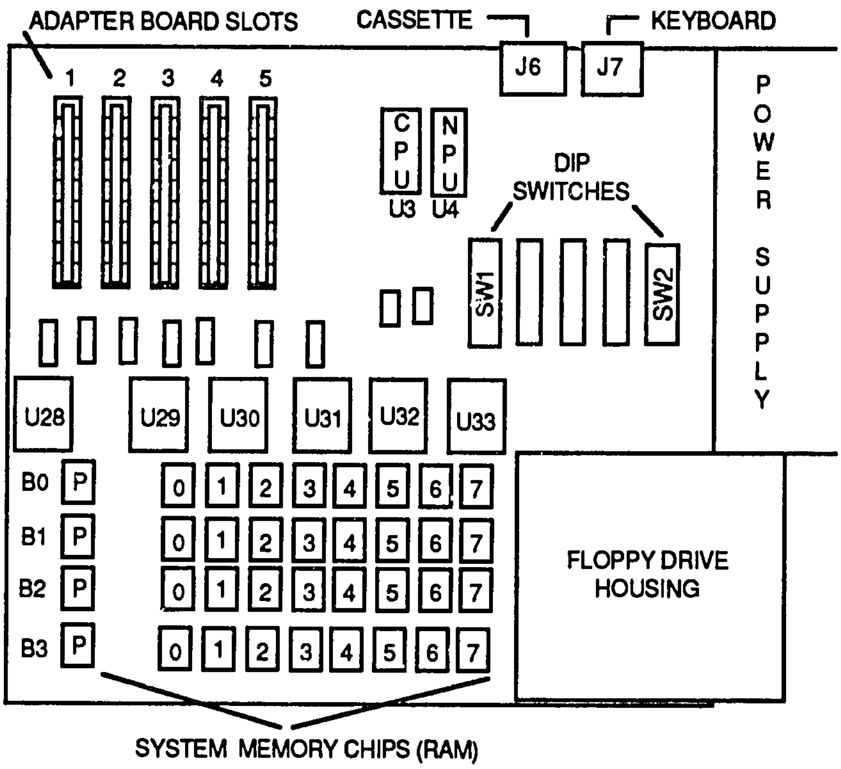


Fig. 3-2. Basic XT System Board Layout

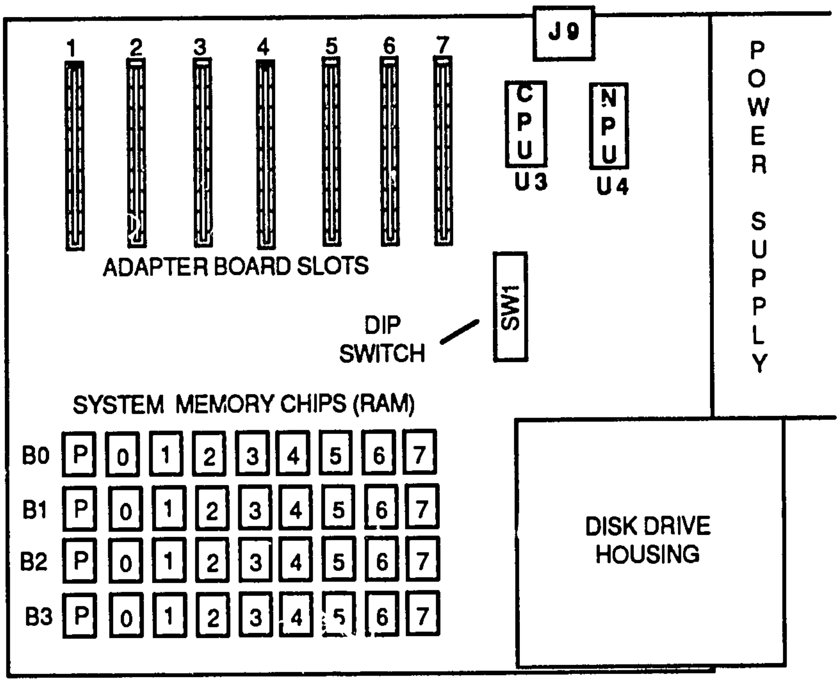


Fig. 3-3. Basic AT System Board Layout

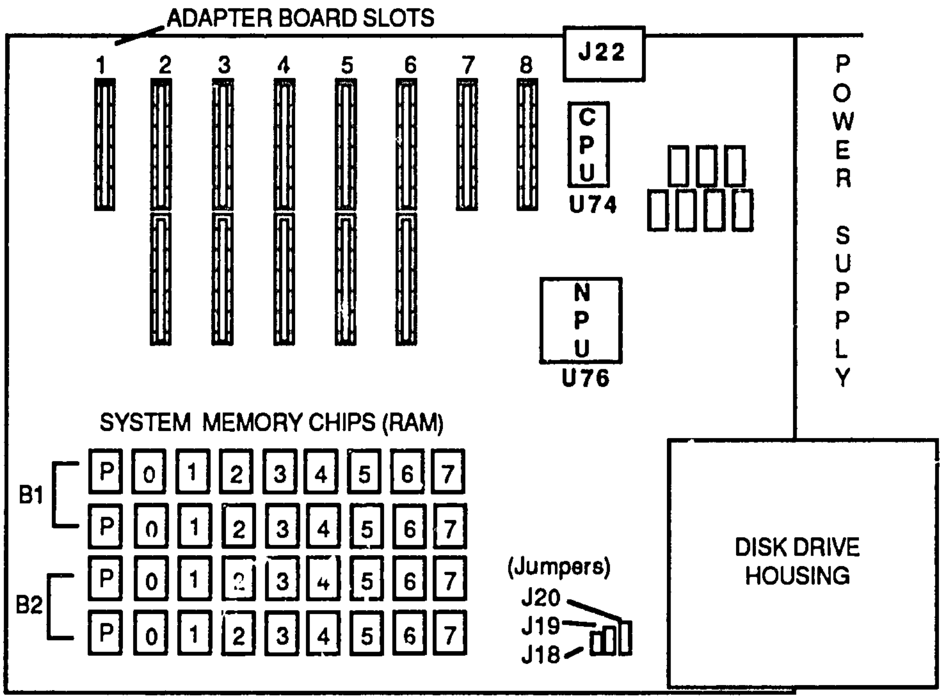
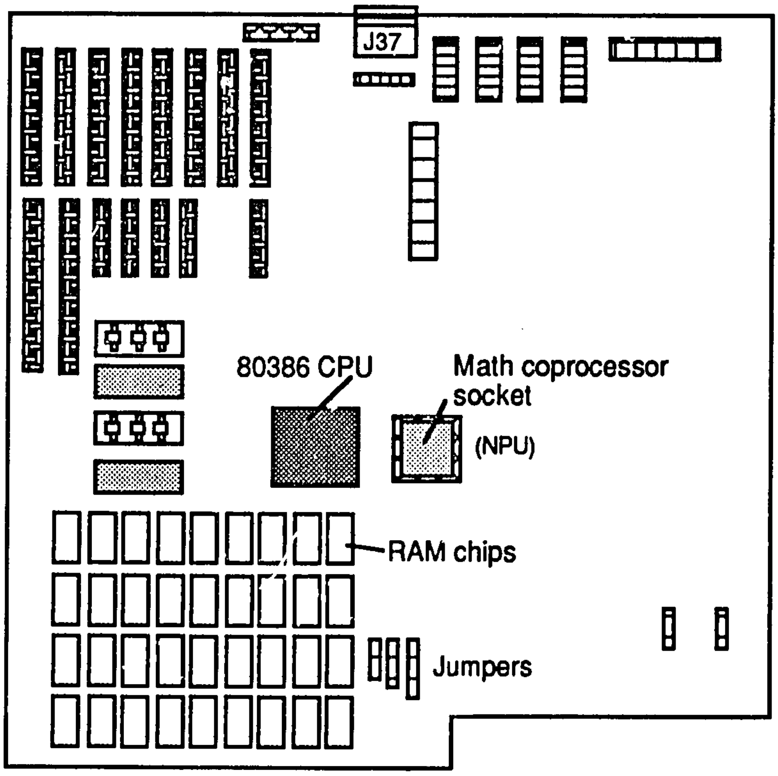


Fig. 3-4. Basic Layout of a Typical 80386 Machine



System Clock Chip

This chip provides the circuitry for using the stream of electrical pulses generated by the system clock (crystal). The crystal and the clock chip determine the maximum speed at which the PC can operate. The actual operating speed, however, is usually less than the maximum afforded by the clock circuit, because the CPU cannot process instructions that fast. Also, other components in the system may be even slower. Therefore, at times, it is necessary for a delay to be initiated until the slower component can execute its functions. These delays are called "wait states."

CPU (Central Processing Unit)

The CPU is a large, complex integrated circuit. Its function is to receive and execute instructions. It is thus called the "brain" of the system. Generally, the speed of the CPU (driven by the system clock crystal and chip) determines the operating speed of the PC. CPU speed is measured in millions of cycles per second. Cycles per second is expressed in "Hertz" units. One million cycles per second is thus expressed as 1 MegaHertz (abbreviated as 1 MHz.) Figure 3-5 shows the speeds for selected models of IBM PCs. At this writing, other PCs (clones) are available that operate at speeds up to 40 MHz.

Despite the ever increasing speed of microprocessor chips, the speed of any system is still limited by the speed of its slowest component. In the world of PCs, the slowest functions are those involving information flow into and out of the system. Therefore, from the standpoint of generating productive output (text, data, graphics), it is important to optimize the data transfer rate to, and output production of, the disk drives and the display. There is little point to increasing the system CPU speed if it still takes all day to paint a new screen, or to retrieve data from a disk.

Math Coprocessor (Numeric Processing Unit, NPU)

The math coprocessor, also called "numeric processing unit (NPU)," is an optional companion to the CPU. It is not necessary for operating the PC. The function of the math coprocessor is to unburden the CPU by performing certain kinds of arithmetic functions, and to perform the operations much faster that the CPU can do them. Math coprocessors use a special method called "floating decimal point," for representing and manipulating numbers. This method allows very large numbers (or numbers that require accuracy out to many decimal places) to be processed quickly. The coprocessor, however, is not activated for doing basic arithmetic or simple calculations, even if they are in a spreadsheet application. The coprocessor specializes in trigonometric and scientific equations, and computations necessary for displaying graphics. Unless these functions are needed in your work, and unless the software you are running can take advantage of a coprocessor, the coprocessor will not be used, even though it is installed. If you do install a coprocessor, be sure to configure the system so the PC will recognize and use it.

Fig. 3-5. Speeds of Various Personal Computers

Type of PC	Speed in MegaHertz (MHz)
PC	4.77
XT	4.77
AT	6 or 8
PS/2 Models 25 and 30	8
PS/2 Model 30-286	10
PS/2 Models 50, 50Z and 80	10
PS/2 Model 55	16
PS/2 Model 70	16 - 25
PS/2 Model 80	16 - 20

Buses

A bus is an electrical path, or collection of paths, that can be shared (plugged into) by two or more components. A telephone system, with its many lines available to be shared by the telephones in offices and residences, is a bus – you simply "plug" your telephones, FAX machines and modems into it. The AC (alternating current) wiring in your home is a bus – all manner of appliances can be plugged into it and draw upon the power coming into the house.

From these two examples, we see that a bus may provide a path for transmitting information or providing a source of electrical power. In many cases, both functions are provided by different lines (e.g., wires or printed circuit paths) within the same bus.

The basic concept of a bus is simple, and the variety of possible designs for any bus is endless. But if every manufacturer used a different design, it would be difficult, if not impossible, to obtain from a third party vendor devices that could be plugged into it. For example, if every local telephone company had a different bus design and each design required a particular design for the instruments that attached to it, the overall inter-city and interstate network would be a mixture of incompatible equipment. It would be expensive to use, and inefficient to operate and maintain.

Eventually, standardization takes place in most major equipment markets. More often than not, it happens because a giant vendor, such as IBM, creates or adopts a particular design, and then everybody else follows suit. Alternatively, several manufacturers may ally themselves to produce a design different from, say, IBM's design (which may be proprietary), and thereby attempt to retain a large share of the market. This is precisely what has happened in the world of the PC.

Regarding PCs, the concept of a bus extends to the notion of a standard – a particular design that many manufacturers use (by mutual agreement or as a matter of economic survival), which will accommodate a host of different plug-in devices made by themselves or by a third party. Of course, a bus must be designed to accommodate the components it serves, and those components, especially the microprocessor (CPU) chips, continue to evolve.

So, attempts at bus standardization are constantly faced with the need to accommodate newer, faster and more powerful components. The early PCs, for example, used an 8 bit CPU chip, requiring 8 data lines. (Note that many lines are needed besides those for data transmission.) Then came the 16 bit chips (needing 16 data lines), and now the 32 bit chips are here. If a chip can send and receive 16 bits or 32 bits at a time (think of it as sending 16 or 32 cars side by side down a highway), then the highway needs a corresponding number of lanes for the cars to travel on. If the highway (bus) is only 8 lanes (bits) wide, for example, it is easy to see that the speed and power of the 16 and 32 bit processors cannot be fully used. Although 32 cars (bits) may be ready to go, they must wait their turn, with only 8 (or perhaps 16) at a time allowed to depart down the road. All the cars may eventually arrive at their destination, but what a slow and wasteful process it would be (and in many cases, is).

In the evolution of PCs, we thus far have seen not one but five or more such "standards." The principle "IBM" buses (excludes the domain of the Macintosh) are as follows: PC, AT/ISA, Micro Channel, and EISA.

PC Bus

The original personal computer bus (the S-100 bus) was that of the Altair computer. It was the de facto industry standard for a long time. Then IBM came along with its PC, and later the XT, and a bus consisting of 62 lines, of which 8 lines carried data. This was fine for the 8 bit, 8088 CPU, but inadequate for the next generation of CPUs, the '286s (i.e., the 80286).

AT/ISA Bus

The next major advance was to the 80286 processor, a 16 bit chip, and IBM's AT class machine. The old PC bus just would not do. It would function, but it was too slow. Eight-bit slots were provided in the AT for backward compatibility – to accommodate existing 8-bit adapter boards. And, 8 bit cards will work in the 16 bit slots, but this wastes the remaining 8 bits of capacity.

IBM's solution was to add another 62 line connector to the existing 62 lines of the 8 bit, PC bus. And so, the 16 bit AT bus was born, complete with 16 bit expansion (adapter) slots. As IBM goes, so goes the PC industry: the AT bus subsequently became known as the Industry Standard Architecture bus or, for short, the "ISA" bus. But it would not be the standard for long.

MCA (Micro Channel Architecture) Bus

Now that the industry "finally" had a standard, the ISA bus, along came the next generation of microprocessor chips – the 386 and the 486. These chips have a 32 bit path, but the old ISA highway offers only 16 lanes. In 1987 IBM went off on its own and came out with its PS/2 family of personal computers. The PS/2s are based on an entirely new architecture (bus design), the so called "Micro Channel Architecture (MCA)," or PS/2 bus. This bus is totally incompatible with the AT/ISA expansion slot format. PC, XT and AT adapter boards cards will not work in it.

Note that this does not apply to all PS/2 models: PS/2 Models 25, 25-286, 30, and 30-286 all support the ISA bus. PS/2 Models 50 through 80 have the MCA bus.

But the news is not all bad. Most peripheral devices – printers, modems, displays – will work with the MCA machines, provided the correct adapter card is installed. And, the need for certain new adapters is eliminated because MCA has video, parallel connectors, and serial connectors built into it.

The MCA bus, however, is proprietary to IBM, and the cost of purchasing the right to clone it is considered by many in the industry to be prohibitive. The rest of the industry had two choices: either acquire the right to clone MCA, or develop something else to handle the 32 bit chips. For the present a least, "something else" appears to be the option selected – the Extended Industry Standard Architecture (EISA), as described below.

EISA Bus

The Extended Industry Standard Architecture (EISA) bus is evolving as a potential major competitor to IBM's MCA bus. The EISA bus is designed for the new, 32 bit

45

chips, having a 32 bit data bus and support for multiprocessing. The main difference between EISA and MCA is that EISA is backward-compatible: the older, 8- and 16-bit ISA adapters will operate in a 32 bit EISA slot.

This contender for the industry standard is being promoted by a consortium of nine substantial companies: AST, Compaq, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wyse, and Zenith. The group is known as the "Gang of Nine." Compaq, however, is in the lead position concerning development and legal rights to the EISA. For this reason, the question remains whether EISA will evolve into an open market for third party vendors, or whether such expansion will be curtailed by Compaq.

In any event, and in contrast to MCA, EISA offers excellent backward compatibility with ISA components. ISA adapters will fit and will run in an EISA system, and the peripheral devices (display, printer, modem, disk drive, etc.) will operate as usual (see "caution," below). The EISA adapter slot has all of the necessary ISA connections in addition to those needed for EISA functions. When an ISA card is inserted, however, special keys obstruct it, so that it will not go all the way into the slot. This ensures that the ISA card will contact the necessary ISA pins, but not the EISA pins in the slot.

CAUTION: **Do not insert EISA cards into ISA system expansion slots. Although the EISA card will fit into the ISA slot, it will not work, and it may prevent the ISA machine from operating. Damage to the card or the system is possible.**

Expansion Slots

All devices that plug into a PC (keyboard, display, printer, disk drive, etc.) require a way to communicate with the PC software. This is accomplished by an "interface" unit that translates information and converts electrical signals between the PC and the peripheral device. Depending on the type of peripheral device, the interface unit is referred to variously as an "adapter," "controller," or "port." The interface unit may take the form of a card (a printed circuit board with electronic components mounted on it), or it may be in the form of a ROM chip (such as that seen for the keyboard "adapter").

The expansion slots hold the "interface" (adaptor, controller, port) cards. Since these slots provide the plug-in cards access to the system bus, they are also referred to as "bus slots."

Generally speaking, a card designed for a given system architecture (e.g., PC, AT/ISA, MCA,EISA) may be plugged into any slot that will physically accept it. There is, for example, no specific slot designated for the disk adapter, video card, multi-function card, or others. Take care not to insert cards from one type of architecture into the slots of another. ISA cards, however, will work in an EISA system, but EISA cards will not work in an ISA system (even though they can be plugged in). Swapping cards by trial and error will likely cause problems, including possible damage to the card, the system, or both.

Memory

In general usage, the word "memory" is defined as the capacity to receive, retain and recall information. In a PC, the term memory refers specifically to a semiconductor device (a "chip") whose main function is to store information. The memory chip, like a blank piece of paper, can be written on. And, like the writing on a piece of paper, the information can be retrieved only by reading it.

In computer parlance, one writes information "to" a chip and reads information "from" a chip. Confusion arises because information also can be stored on and retrieved from other devices, such as disks, magnetic tape and optical media. These devices are properly termed "storage" devices, if for no reason other than to eliminate unnecessary confusion. Careless use of language produces statements such as, "My hard drive has 80 megabytes of memory." The concepts and technology concerning memory and storage are difficult enough, without thus compounding the problem. So, unless the "memory" in question resides on a chip, consider it "storage capacity," or use any name for it other than "memory."

In PCs, there are several different kinds of memory chips, each having a particular class of function:

- Random Access Memory (RAM) chips
- Read-Only Memory (ROM) chips
- Complementary Metal Oxide Semiconductor (CMOS) chips

Random Access Memory (RAM)

RAM chips can be both written to, and read from, by the CPU. These chips are used for storage and retrieval of program instructions (computer code) and data when the PC is powered up and operated. By loose analogy, RAM memory is like the "short term" memory of the human brain. Its primary function is to support immediate, ongoing operations and calculations performed by the CPU – just as in doing your daily work your brain's short term memory stores your "do list," digits

47.

and subtotals when you do math problems, telephone numbers you have just looked up in order to dial, and short bits conversation.

RAM chips are "high speed" chips. They receive and release information at a very high speed, much faster than other types of chips in the PC. When the power is turned off, however, all of the information in the RAM chips disappears. RAM memory is thus referred to as "volatile" memory – it "evaporates" in the absence of electricity to support it. The function performed by RAM chips could theoretically be done by a disk, but the process would be so slow as to be totally impractical.

Read-Only Memory (ROM)

ROM chips cannot be written to by the CPU. They can only be read by the CPU. ROM chips contain software instructions and information written to them by the manufacturer, using special equipment. The Basic Input-Output System (BIOS) for the PC is stored on a ROM chip on the motherboard. The BIOS is a software program by which the CPU communicates with the PC's hardware. When the PC is booted up, the information stored on the ROM BIOS chip is copied (read) into RAM memory, from whence it is used by the CPU during a work session. ROM chips may also be found on expansion boards (adapter cards).

Unlike a RAM chip, a ROM chip does not need a supply of electricity (either from the PC power bus, or from a battery) in order to retain its information. When the PC power is turned off, the ROM chip still "remembers."

Complementary Metal-Oxide Semiconductor (CMOS) Memory

CMOS (Complementary Metal-Oxide Semiconductor) chips are just another form of RAM chip. They can be both written to and read by the CPU, and they need a source of electricity in order to retain their information. They are used commonly to store information about the PC's configuration, e.g., the amount and address of RAM installed; hardware options installed; the disk drive parameters; Direct Memory Access (DMA) and Interrupt Request (IRQ channels); and so on. (DMAs and IRQs are covered in Chapter 11.)

In such application, the CMOS chip is supported by a small battery. The battery allows the chip to retain its information. It is independent of the PC power supply, however: In the absence of the battery, the CMOS chip does not use power from the PC. If the battery dies or is removed, the information is lost within a short time.

In summary:

RAM = high speed, working memory; CPU can write to it and read from it; it is the CPU's "short term memory bank." RAM needs electrical power in order to retain its information.

ROM = relatively "permanent" memory in that it does not require electrical power in order to retain its information; CPU can read information from ROM, but cannot write to it; stores the system BIOS on the motherboard; stores programs on adapter cards (e.g., graphics boards, video boards).

CMOS = another form of RAM; CPU can write to it and read from it; when used to store configuration information, CMOS requires a supply of current from a battery, to retain its information, regardless of whether PC power is on or off.

Let's now see how all of this memory is organized and used.

Memory Areas and Memory Mapping

The main memory on a PC (also called "planar memory") is organized into various areas: conventional (user) memory, reserved area, extended memory, and expanded memory. Certain of these areas are further divided into functional sub-areas. The names given these areas are often less than illuminating, and are frequently confusing, especially as the same area may be called by different names. Below, we will define the main terms and concepts. As we go along, we will construct a memory map, step-by-step.

Conventional Memory

The first 1Mb (1024Kb) "chunk" of RAM on the PC is divided into two successive areas, 640Kb and 384Kb. The first area (640Kb) is the "conventional memory area." The remaining 384Kb is "reserved," and is thus not available to DOS (see Reserved Memory Area, below). The conventional memory area is immediately available for loading and running applications such as word processing, spreadsheets, etc. Accordingly, this area is also known as "user memory." When the system is booted up, DOS is loaded into this area, as are any TSR (terminate and stay resident) files to be used, such as device drivers; screen "savers" (blankers); Borland's Sidekick; and utilities such as DOSKEY. The space remaining after these files are loaded is what is actually available for your application programs, documents spreadsheets, etc. Figure 3-6 shows the first part of the memory map. The figure will be expanded upward, adding the other memory areas as we go along.

Fig. 3-6. Conventional (User) Memory Portion of PC Memory Map

Absolute Decimal Address Range	Memory Area
0 to 640K-1 (Note that counting begins at zero)	Conventional Memory (640K) - DOS, TSRs, Drivers - Applications programs/files

There is a common misconception that DOS can address only the first 640K of memory. In fact, DOS could address the entire 1Mb, except that the remaining 384K is reserved for other use. Since DOS can load applications programs only into contiguous memory areas (i.e., memory addresses that are in sequence, with no gaps or jumps), the remaining 384K of DOS's addressing capability cannot be used for applications. The next higher, contiguous memory is the reserved memory area.

Reserved Memory Area

The first 1Mb (1024K) of RAM on the PC is divided into two areas. The first 640K is the "conventional memory" area, as explained above. The remaining 384K is the "reserved area." This area is also known as the "upper memory block" and the "high memory area." Of the reserved memory area, 128K is set aside for use by the video (monitor) to keep track of characters and graphics to be displayed on the screen. This 128K is called the "video memory area."

Most of the remaining 256K of the reserved memory area is for use by certain expansion cards such as LAN (Local Area Network), VGA (Video Graphics Array), and EGA (Enhanced Graphics Adapter) boards. The boards may require this space, for example, to temporarily store (buffer) information, or to map the software stored in their ROM chips. Also located in the reserved area is a 64K buffer (four, 16K "pages"). This buffer acts as an access window to the expanded memory area (to be described shortly). Figure 3-7 shows the conventional memory area and the reserved memory area.

Fig. 3-7. First Megabyte of Memory is Divided into 640Kb of Conventional (User) Memory and 384Kb of Reserved Memory

Decimal Address Range	Memory Area
640K to 1024K-1	Reserved Memory Area (384K) - EMS window (64K) - Buffers and ROM - Video Memory (128K)
0 to 640K-1	Conventional Memory (640K) - DOS, TSRs, Drivers - Applications programs/files

Video Memory

Video memory is 128K of memory, reserved for use by the video (display monitor) to keep track of text and graphics for display on the screen. The video memory area is a sub-area of the reserved memory area. The reserved memory area consists of the 384K of RAM, just above the 640K, conventional (user) memory area.

Extended Memory Area

Extended memory is that memory space (range of addresses) which begins immediately above the first 1024K of memory. This area can be used by Xenix (operating system), OS/2 (operating system), a few of the DOS programs (but not by DOS as an operating system), and certain applications programs (e.g., Lotus 1-2-3, Version 3.0 or higher.

Recall that the first 1024K (i.e., the first 1Mb of RAM) is occupied by the conventional memory (640K) area plus the reserved memory area (384K). The lower boundary of the extended memory is immediately above the 1024K. The upper boundary of the extended memory area (i.e., the size of this area) depends on the CPU in the machine. For 80286 machines, it is (up to) 16 Mb. For 80386 and 80486 machines, it is (up to) 4096 Mb. Machines using the 8088 chip (the PC and XT), and PS/2 Models 25 and 30, which use the 8086 chip, cannot have extended memory. Figure 3-8 shows the conventional, reserved and extended memory areas.

Fig. 3-8. Conventional, Reserved and Extended Memory Areas

Decimal Address Range	Memory Area
1024K to (max of) 16Mb for 386 and 486 machines. 1024K to (max of) 16Mb for 286 machines.	Extended Memory Area (Not available for PC, XT, or PS/2 Models 25 and 30)
640K to 1024K-1	Reserved Memory Area (384K) - EMS window (64K) to expanded memory area -------> - Buffers and ROM - Video Memory (128K)
0 to 640K-1	Conventional Memory (640K) - DOS, TSRs, Drivers - Applications programs/files

Expanded Memory

Since we have been building the memory map, adding each new area on top of the last, logic might suggest that the "expanded memory area" would now be added atop what we have built so far. That assumption would be incorrect. Expanded memory is, in fact, just additional RAM that can be accessed only in a special way. Expanded memory can be as large as 32Mb. However, expanded memory is totally out of the range of the memory that can be addressed by the CPU. It cannot be used to run applications programs, but data from applications can be stored in it. This assumes that the application being used can take advantage of the available expanded memory.

The only way to get data into and out of expanded memory is through a small "window" (buffer) located in the reserved memory area. This window can be addressed by the CPU, and thus data can be sent back and forth through the window. To feed information into the expanded memory area, it must be divided into "pages" of 16K each. Expanded memory does not have addresses of the kind discussed for the other memory areas. Instead, it uses page numbers, i.e., information is fed into it (and retrieved) page by page, and each page is numbered.

The window buffer just mentioned consists of four, 16K pages, for a total of 64K. All four pages can be active (used) at one time. Thus, if you had 640K of data to feed into the extended memory area, it would have to go through the 64K window in 40 pieces (pages) of 16K each. The same applies to getting the data back out again. This narrow window of access slows things down, but the additional RAM is nonetheless useful. Certain DOS applications take advantage of this area. Among them are Lotus 1-2-3, Version 2.0 and higher; Framework; SuperCalc; and Memory Mate. Thus, it allows you to create, for example, a spreadsheet file that is larger than would otherwise be possible using just the conventional memory area. Figure 3-9 shows the conventional, reserved, extended and expanded memory areas.

Upper Memory Block

This is another name for the reserved memory area. It is the memory located between the upper boundary of the conventional memory area and the lower boundary of the extended memory area.

High Memory Area

This is another name for the reserved memory area. It is the memory located between the upper boundary of the conventional memory area and the lower boundary of the extended memory area.

Expansion Memory

Expansion memory is a general term. It refers to any RAM added to the PC to expand its capacity in one area or another. For example, if you have 256K of RAM installed in your conventional (user) memory area and you now purchase some RAM chips to increase the size of your conventional memory area, you are purchasing "expansion memory." The same applies to purchasing and installing RAM chips for your extended memory and expanded memory areas. Expansion memory increases memory somewhere on the machine. But by itself, the term does not indicate where the new memory is to be installed.

Color Burst Adjustment

This adjustment is used when the PC is connected to a color television set.

Fig. 3-9. Memory Map of a Personal Computer

Decimal Address Range	Memory Area	
1024K to (max of) 16Mb for 386 and 486 machines. 1024K to (max of) 16Mb for 286 machines.	Extended Memory Area (Not available for PC, XT and PS/2 Models 25 and 30.)	
640K to 1024K-1	Reserved Memory Area (384K) - EMS window (64K) to expanded memory area ---> - Buffers and ROM - Video Memory (128K)	EXPANDED MEMORY Up to 32 Mb
0 to 640K-1 (Note that counting begins at zero. Therefore, the upper number is 640K minus 1)	Conventional Memory (640K) - DOS, TSRs, Drivers - Applications programs/files	

Keyboard Adapter

Peripheral devices require some form of interface (adapter) to communicate with the PC, and the keyboard is no exception. Whereas most adapters for peripherals are in the form of a card (board) that plugs into an expansion slot, the keyboard interface (adapter or controller) is (usually) in the form of a ROM chip mounted on the motherboard. This is commonly referred to as the "keyboard ROM."

Keyboard

The keyboard is the most common input device for a PC. The keyboard cable connects to a receptacle at the back of the motherboard. Or, if an adapter card is used, the keyboard cable connects to the card.

Power Supply

The power supply receives AC (alternating current) from the power cord that connects the PC to a wall outlet. The power supply converts then AC power to DC (direct current) power to operate the PC cooling fan, disk drives and computer logic/memory functions.

Display (Monitor) and Display Adapter

Most commonly, PCs use a display based on Cathode Ray Tube (CRT) technology. In brief, a CRT is a glass tube that has been sealed and evacuated (the air has been pumped out of it). The display surface (screen) is coated with phosphor. Inside the tube, an electron gun shoots electrons against the screen, which causes the phosphor particles to glow and thus to produce a visible display. By using a combination of phosphors that emit different colors, it is possible to produce colored images on the screen.

The PC, XT and AT family of machines require an interface (video adapter) card, which occupies one of the expansion slots on the motherboard. These include the familiar MDA (Monochrome Display Adapter), CGA (Color Graphics Adapter), Enhanced Graphics Adapter (EGA). Note that in certain models of the PS/2 family, the video adapter is built directly into the motherboard.

Floppy Disk Drive and Controller

Being a peripheral, a floppy disk drive requires an interface card (controller) to communicate with the computer. The card fits into one of the expansion slots on the motherboard. A single board may contain the circuitry to support both a floppy drive and a hard drive. Note that XT controller boards and AT controller boards are not interchangeable.

Hard (Fixed) Drive and Drive Controller

A hard disk drive requires an interface card (controller) to communicate with the computer. The card fits into one of the expansion slots on the motherboard. A single board may contain the circuitry to support both a hard drive and a floppy drive. Note that XT controller boards and AT controller boards are not interchangeable.

Multi-function Board

As its name suggests, a multi-function board combines certain capabilities that would otherwise be located on separate boards. A common type of multi-function board contains the following:

- Printer Port: the interface between the computer and the printer. Printer ports are most commonly "parallel" ports. A parallel port transports 8 data bits at a time (simultaneously), rather than one bit at a time, as with a serial port.

- Clock Calendar: a timing circuit that, by virtue of a battery, runs continuously to keep track of the calendar date and the time of day. This should not be confused with the computer crystal and clock circuit, whose purpose is to generate the stream of electronic pulses by which the computer coordinates its processing activities.

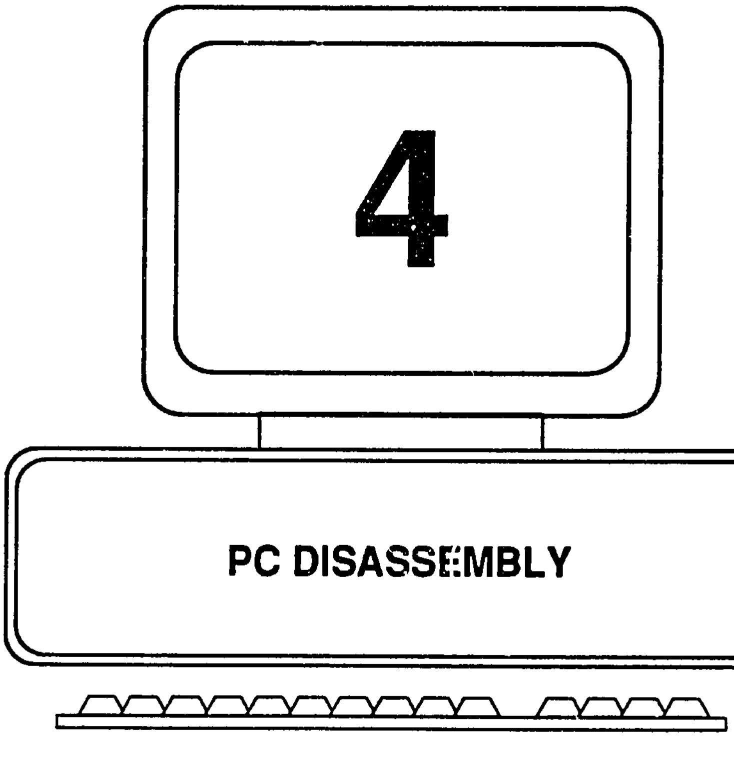
- Serial Port (RS 232C Port): an interface used for serial (as opposed to parallel) communications. This type of port provides output to serial printers and to serial communications devices (e.g., modems).

From the material covered in this chapter it should be clear that a basic PC has only a limited number of primary components, and that each component supports a particular input, processing or output function. In contrast to most office equipment and home appliances, a basic PC usually requires the addition of several peripheral devices and their associated adapter (interface) cards before it begins to approach its full capability.

In most cases, the addition of a peripheral device must include one or more software files called "drivers" (see Chapter 7) in order for the peripheral to function. Further, the system must be configured (i.e., informed of the presence and specifications of the peripheral device, such as a video display, disk drive, printer). Installation of adapter boards is covered in Chapter 12. Disk drive installation is addressed in Chapter 13.

NOTES

51


4
PC DISASSEMBLY

CHAPTER 4: PC DISASSEMBLY

PC, XT, AT, AND PS/2 MODELS

General Guidelines

Before you begin disassembling a PC:

- Be aware of the electrical hazards and plan to follow all safety precautions.

WARNING:

Never attempt to take a power supply or a display monitor apart. These devices retain a strong electrical charge even when they are not plugged into a power source. The shock is highly dangerous. Further, there is nothing inside them that you can repair.

Never attempt to operate a power supply whose output leads have been disconnected from the system it supplies. The power supply will be destroyed, and it may even explode!

Always check the power requirements of a component, especially display monitors and power supplies before swapping or installing them. The AC input power source (voltage and frequency) must match that required by the device.

- Be certain that disassembly is absolutely necessary – have you investigated and ruled out all steps that do not require opening up the machine to solve the problem?

- Discharge static electricity from your body by touching a metal doorknob or a piece of metal furniture.

- Backup the hard drive. This assumes that the hard drive is functioning. If there is a problem with the drive, especially evidence of read-write errors, copy the data from the hard drive. The DOS XCOPY command is faster that the COPY command. Then backup the files. If old backup disks already exist, save them; do not use them. Instead, use a clean set of floppies for backup. This way, you will at least have the old set of backups if the new ones are defective.

- Backup (copied to disk or printed out) the configuration data. Do this by booting the PC, running the SETUP program and printing a copy of the configuration data screen using the PRINT-SCREEN key. Alternatively, run a diagnostic routine such as Checkit, and save the configuration data to a floppy, print out a copy, or preferably do both.

- Park the hard drive heads. If you are working on an XT, you can boot the PC from the XT diagnostics disk and then run SHIPDISK. Do not run SHIPDISK from a floppy other than the diagnostics diskette, or from the hard drive, if SHIPDISK has been copied to the hard drive. Doing so will destroy data on the hard disk. Also, never run an XT copy of SHIPDISK on a AT. This will destroy the data on the AT hard drive. Preferably, use the appropriate reference diskette or a third party utility.

- Place a cardboard shipping insert or a "scratch" diskette into the floppy drive and close the door. A scratch diskette is simply one that has no useful data on it. It does not have to be formatted. This procedure will keep the floppy drive heads from striking each other as you move the machine and work on it.

- Lay out the tools and materials you need. This includes: the basic mechanical tools; labels and tape for marking connectors and adapters; containers for storing and identifying screws to be removed; and writing materials for diagramming component locations and cable connections. Also on hand should be any reference documentation for the machine you are working on. Making your own sketches and diagrams, however, is essential! This should include a record of adapter card slot positions, cable connections, DIP switch and jumper settings, once the PC is opened. Trusting to memory is to invite disaster.

- Allow yourself enough room in which to work. Preferably, move the PC to an uncluttered area, where you can lay out the components in order as they are removed. This will also avoid dropping or misplacing objects and tools.

- Disconnect the power cords and external cable connectors to the PC from all peripherals (printer, monitor, etc.). Be sure to label each corresponding plug and socket before removing them. Set the monitor aside. A safe practice is to place the monitor out of the way, on the floor, with the display side toward the wall.

- Unlock the cover (certain models have a keylock), and remove the key to a safe place, where it won't be lost.

- Plan to work slowly, carefully and methodically. In the long run, you will accomplish more in less time than if you rush. And, you will avoid errors that could make matters worse than they were before you started.

Recommended Set of Tools

The following tools will suffice for disassembling and reassembling personal computers, and for making minor repairs. Note that NO magnetized tools should be used.

- Nut drivers – 3/16 and 1/4 inch
- Screwdrivers
  - Phillips, small and medium
  - Flat blade, small and medium
  - TORX T10 and T15
- Tweezers
- Needle-nose pliers
- Wire cutter/stripper
- Chip Extractor
- Chip Inserter
- Wrap Plug (for testing ports)
- Small Flashlight
- Wrist Strap (for static discharge)

PC and XT Disassembly

Components will be removed in the following order:

1. Cover
2. Adapter Boards
3. Disk Drives
4. Power Supply
5. Motherboard

Tools/materials needed:

- 1/4-inch flat blade screwdriver
- 3/16-inch nut driver (optional; a screwdriver will do)
- Envelopes or containers (e.g., paper cups) for storing screws (one container for each set of screws). Store the screws carefully or you will have a problem replacing them.
- Stick-on labels or paper tape for labeling connectors and receptacles
- Paper and pencil or pen for diagrams and notes

Work slowly, carefully and gently. Never force, twist or bend components. Components will be removed easily if proper procedures are followed.

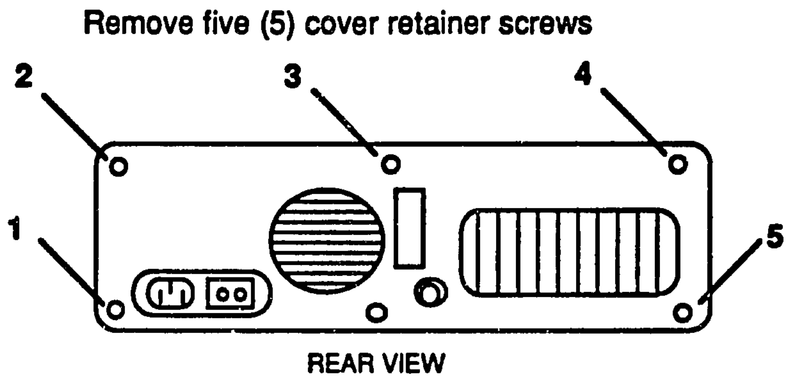
- Label all external cable connectors and receptacles. (Use stick-on labels or paper tape)
- Disconnect all power cords from equipment and from wall outlet
  - Monitor
  - System Unit
  - Printer
  - Other
- Disconnect all remaining external cables and set monitor aside.

You are now ready to begin disassembling the system unit.

Remove Cover

1. Place the system unit on a table or workbench with adequate room to work; position so you are looking at the rear panel (Figure 4-1).

Fig. 4-1. PC/XT Rear Panel



2. At the rear panel, locate five (5) screws as shown in Figure 4-1.

3. Using the nut driver or screwdriver, remove the five (5) screws. Place the screws in container; label container "back panel cover screws." This will ensure that the screws are not lost and that they do not accidentally fall inside the unit, where they may cause a short circuit.

4. Slide the cover assembly forward as far as it will go; then lift the front of the cover up and remove it.

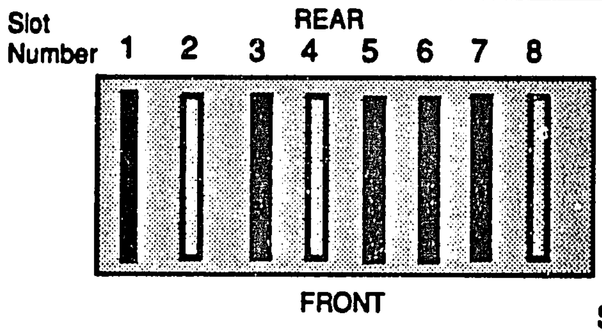
Remove Adapter Boards/Cards

Items already removed: cover

1. Observe and diagram the location and orientation of each adapter card. If you cannot identify a card by its function, mark it and the slot it sits in (use label, paper tape, etc.).

2. Make a sketch of the adapters (as illustrated in Figure 4-2), showing name, location (slot) and orientation of each adapter; show position of each cable (if any) connected to each adapter. (Odd-colored stripe on ribbon connector indicates pin number one (Pin #1). Also show the position of each jumper and switch on your sketch. On the board, switches are labeled "SW" and jumpers are labeled "J."

Fig. 4-2. Example of Format for Adapter Location Sketch



Slot	Adapter	Jumper	Switch settting
1	Video board		0110
2	Empty		
3	Disk drive controller		
4	Empty		
5	Com port		

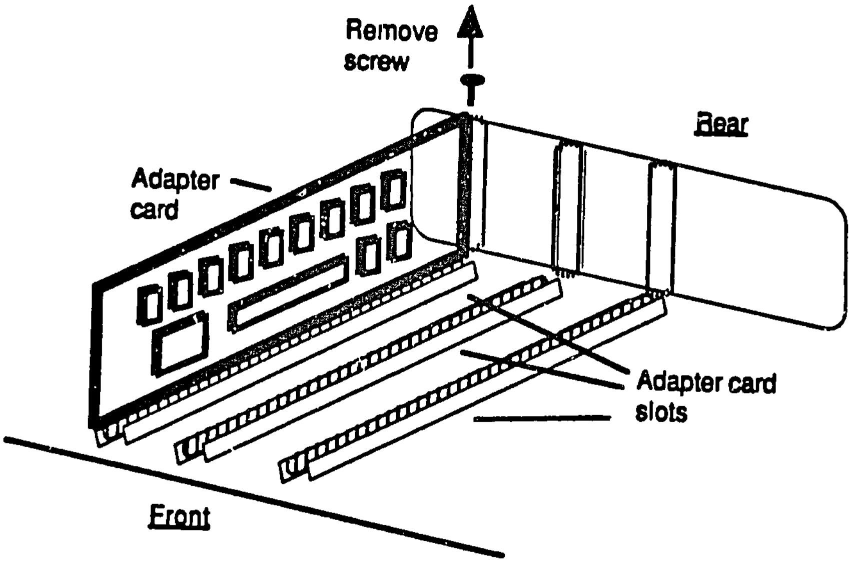
-- etc. --

3. Using the nut driver, remove the retainer screw(s) that hold the adapter(s) in place (Figure 4-3). Place screws in container and label container "adapter retainer."

4. To remove an adapter, grasp the adapter at both ends and lift upward evenly at both ends. Do not twist or bend the adapter.

5. Remove all remaining adapters and set aside. (Arrange them on the table in the same orientation as shown in your sketch; see Figure 4-2: Example of Format for Adapter Location Sketch.)

Fig. 4-3. Location of Adapter Retaining Screw



Remove Disk Drives (Hard Drive and Floppy Drive)

Items already removed: cover; adapter cards

1. Stand the unit on end, with the front panel facing upward.

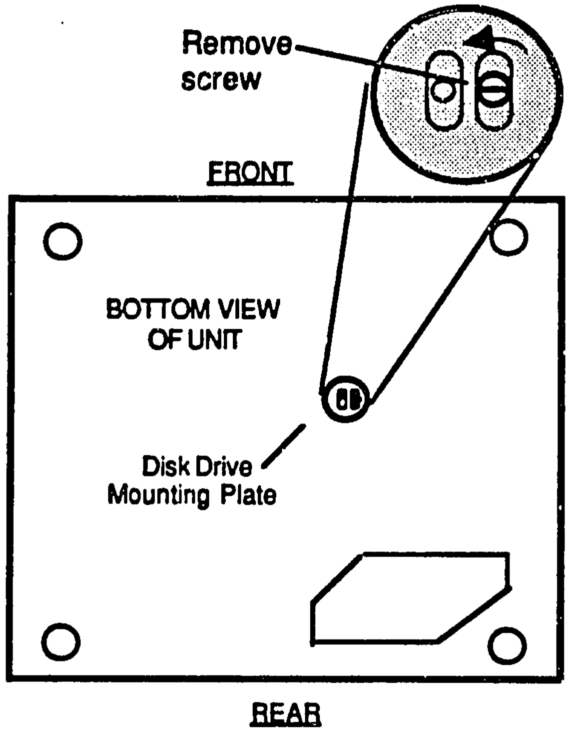
2. Position unit so that you are looking at the bottom of the unit.

3. Locate and remove the hard drive (Drive C) retainer screw at the bottom of the chassis (Figure 4-4).

4. Place screw in container and label as "hard drive; bottom of chassis."

CAUTION: **Do not get the screws mixed up. If you use a screw that is too long, you may damage the system.**

Fig. 4-4. Hard Drive Retainer Screw on Bottom of PC



5. Place the chassis flat on the table. Position it so that you are looking at it from the front.

6. Locate and remove the hard drive mounting screws as shown in Figure 4-5.

7. Place screw in container and label as "hard drive mounting screws."

8. Locate and remove floppy drive mounting screws as shown in Figure 4-6. Place screws in container and label container "floppy drive mounting screws."

66

Fig. 4-5. Hard Drive Mounting Screws

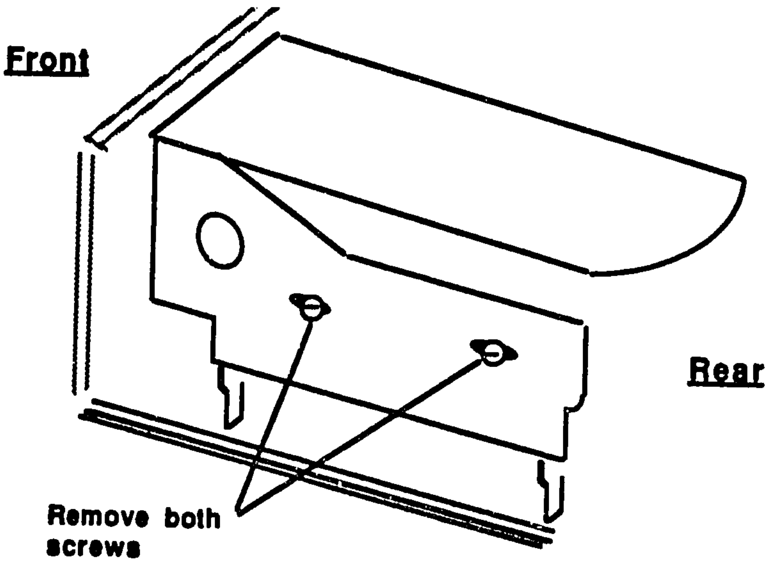
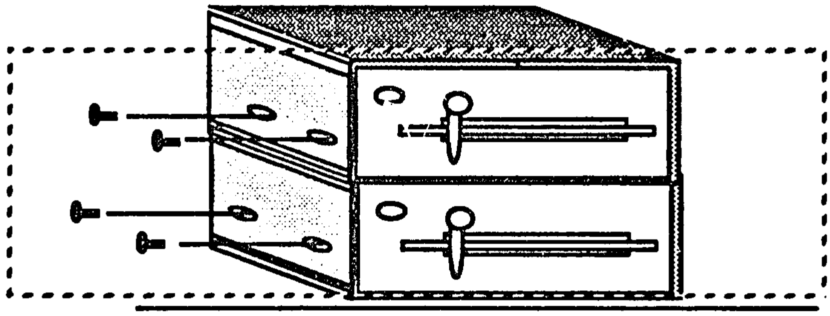


Fig. 4-6. Floppy Drive Mounting Screws



9. Slide the floppy drive forward about two inches, and then disconnect the power connector and signal cable at the top-rear of the drive, as shown in Figure 4-7(a).

CAUTION: Do not pull on wires or cables. Grasp and pull on the connectors.

10. Slide the floppy drive all the way out and set it aside.

11. Slide the hard drive forward about two inches.

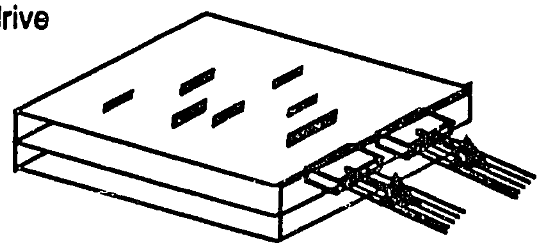
12. At the lower rear of the hard drive (Figure 4-7(b)), label and disconnect:

- Data cable
- Control signal cable
- Power cable
- Ground lead

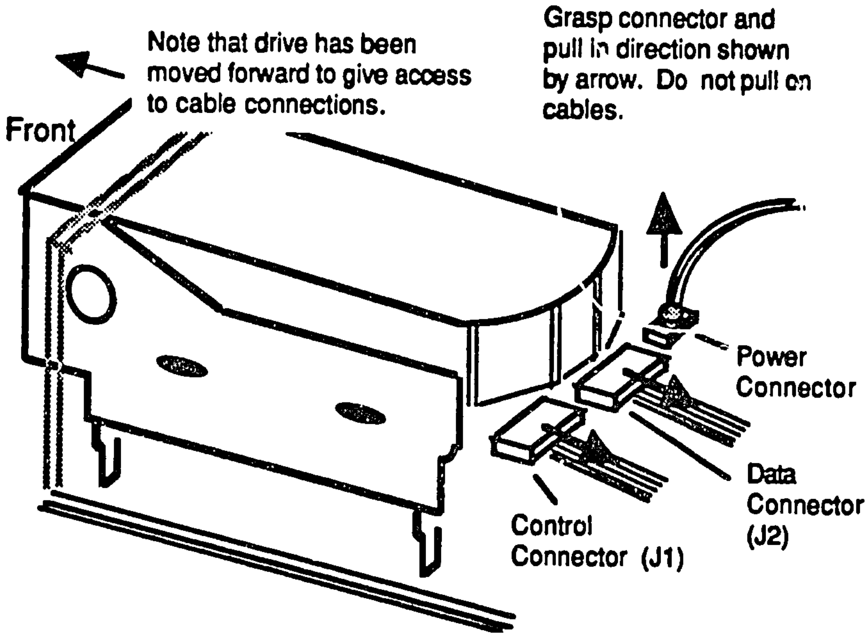
13. Slide the hard drive out and set it aside.

Fig. 4-7(a) and (b). Floppy Drive and Hard Drive Cable Connections

(a) Floppy drive



(b) Hard drive



Remove Power Supply

Items already removed: cover; adapter cards; floppy drive; hard drive

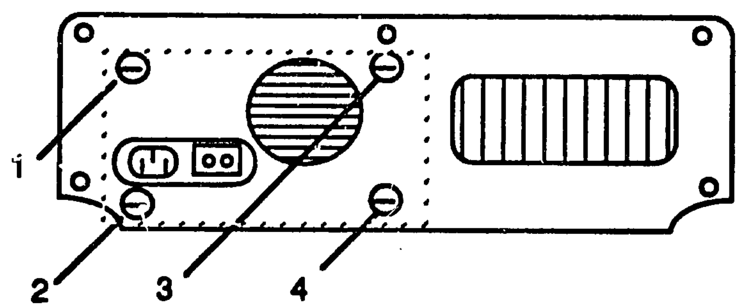
Note: To remove the power supply, four mounting screws must be removed and the power supply must be slid forward to disengage the retaining tabs beneath it.

1. At the rear of the chassis, locate and remove the four (4) power supply retaining screws (Figure 4-8).
2. Label and disconnect the cables between the power supply and the motherboard (Figure 4-9).

CAUTION: Do not pull on wires or cables. Grasp and pull on the connectors.

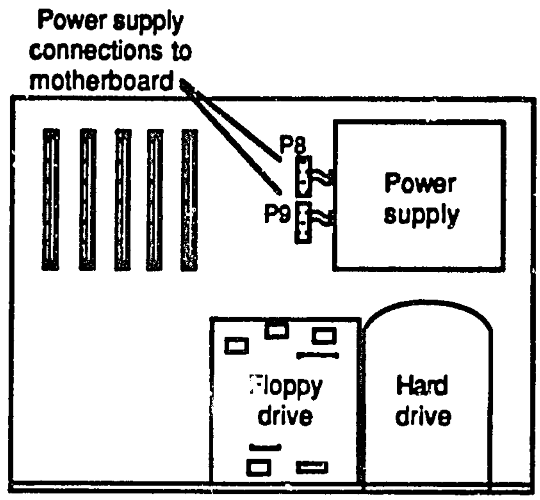
Fig. 4-8. Power Supply Retaining Screws

Remove four (4) power supply mounting screws



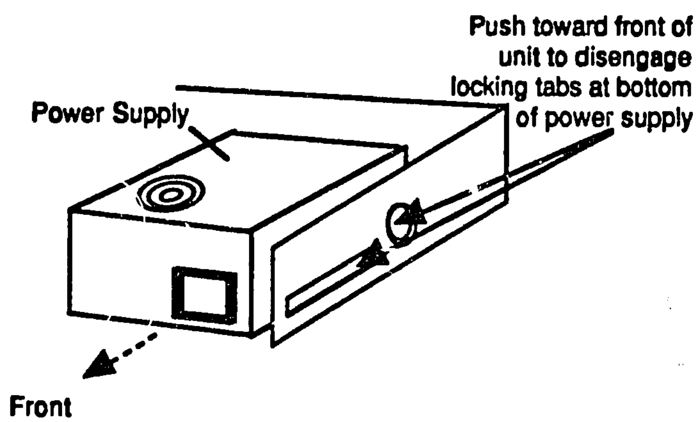
Rear view of system (back panel)

Fig. 4-9. Power Supply Cable to Motherboard



3. If not already disconnected, disconnect the cables between the power supply and the disk drives.
4. Slide power supply outward to disengage tabs beneath it (Figure 4-10).
5. Lift out the power supply and set it aside.

Fig. 4-10. Disengage Power Supply Retaining Tabs



Remove Motherboard

Items already removed: cover; adapter cards; floppy drive; hard drive; power supply

The motherboard is held in place by retaining screws and by plastic "spacers" or "standoffs." The standoffs hold the motherboard away from the metal chassis to prevent short circuits. The standoffs fit into slots. After removing the retaining screws, the board must be slid sideways to disengage the standoffs from their slots.

1. Make a sketch of the motherboard, showing position of all cables attached to it.

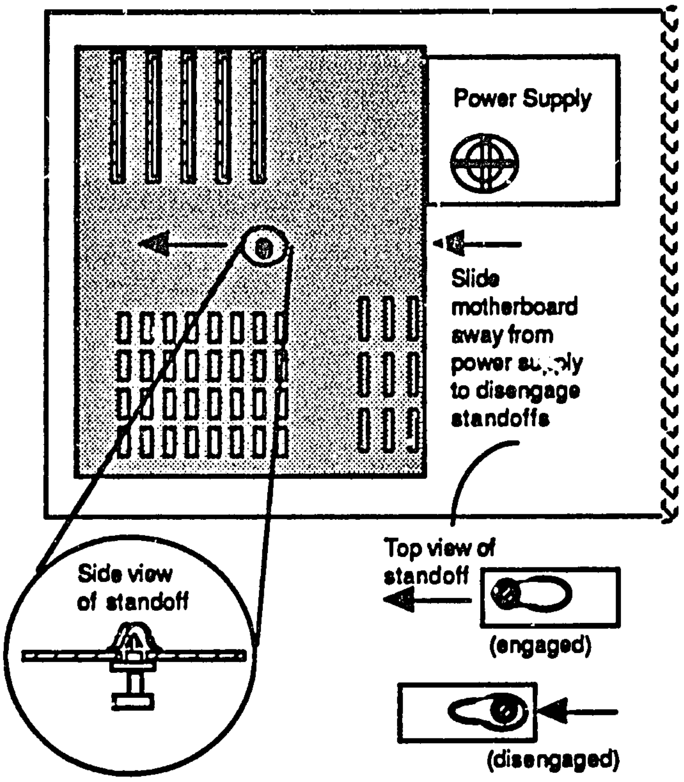
2. Label and disconnect all cables to the motherboard.

3. Remove the retaining screws; place the screws in a container and label the container "motherboard retaining screws."

4. Disengage the standoffs by sliding the motherboard in the direction away from the power supply location (Figure 4-11). Be sure that the standoffs have been disengaged before going to next step.

5. Grasp the opposite edges of the motherboard and lift the board straight up and out.

Fig. 4-11. Disengage Motherboard Standoffs



CAUTION: **Do not force or flex the board. If it does not come out easily, check that you have removed all of the retaining screws and that the standoffs have been disengaged.**

AT Disassembly

The procedures here are essentially the same as for the PC/XT models. For the AT, however, the disk drives are mounted on plastic rails. The drives slide in and out on these rails. Also, the drives are held in place by metal tabs (and/or a "keeper bar") at the front of the drive.

Components are to be removed in the following order:

1. Cover
2. Adapter Boards
3. Disk Drives
4. Power Supply
5. Motherboard

Tools/materials needed:

- 1/4-inch flat blade screwdriver
- 3/16-inch nut driver (optional)
- Envelopes or containers (e.g., paper cups) for storing screws (one container for each set of screws). Store the screws carefully or you will have a problem replacing them.
- Stick-on labels or paper tape for labeling connectors and receptacles
- Paper and pencil or pen for diagrams and notes

Work slowly, carefully and gently. Never force, twist or bend components. Components will be removed easily if proper procedures are followed.

- Label cable connectors and receptacles. (Use stick-on labels or tape)
- Label and disconnect all power cords from equipment and wall outlet
  - Monitor
  - System Unit
  - Printer
  - Other
- Label and disconnect all external cables and set monitor aside.

Remove Cover

1. Place the system unit on a table or workbench with adequate room to work; position it so you are looking at the rear panel.

2. At rear panel, locate 5 screws as shown in Figure 4-12.

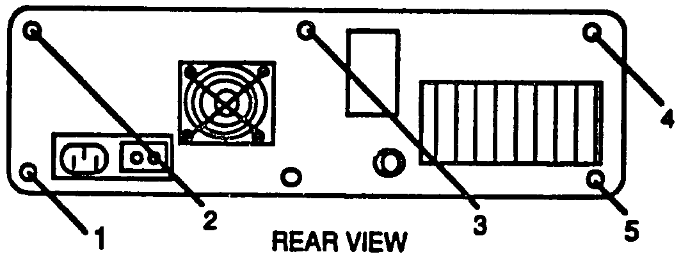
3. Using nut driver, remove the five (5) screws.

4. Place screws in container; label container "back panel cover screws." This will ensure that the screws are not lost and that they do not accidentally fall inside the unit, where they may cause a short circuit.

5. Slide the cover assembly forward as far as it will go; then lift the front of the cover up and remove it from the unit (chassis).

Fig. 4-12. AT Cover Screws on Rear Panel

Remove five (5) cover retainer screws



Remove Adapter Boards/Cards

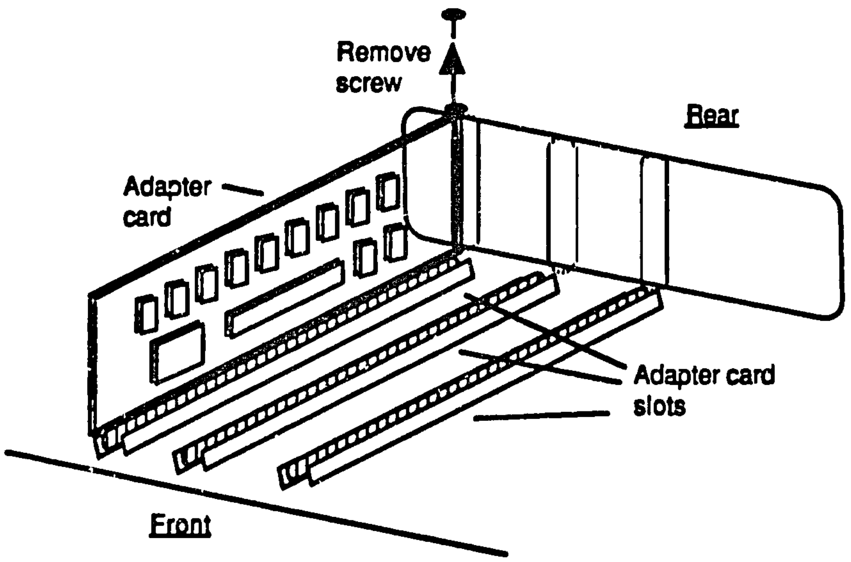
Items already removed: cover

1. Observe and diagram the location and orientation of each adapter card. If you cannot identify a card by its function, mark it and the slot it sits in. (Use label, paper tape, etc.)

2. Make a sketch of the adapters showing name, location (slot) and orientation of each adapter; show position of each cable (if any) connected to each adapter. (Odd-colored stripe on ribbon connector indicates pin number one (Pin #1). Also show the position of each jumper and switch on your sketch. On the board, switches are labeled "SW" and jumpers are labeled "J."

3. Using the nut driver, remove the retainer screw(s) that hold the adapter(s) in place (Figure 4-13). Place screws in container and label container "adapter retainer."

Fig. 4-13. Adapter Card Retaining Screw



4. To remove an adapter, grasp the adapter at both ends and lift upward evenly at both ends. Do not twist or bend the adapter.

5. Remove all remaining adapters and set aside. Arrange them on the table in the same orientation as shown in your sketch made during disassembly.

Remove Disk Drives

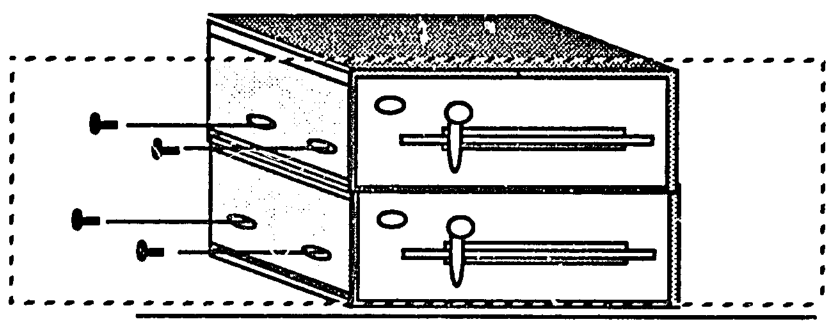
Items already removed: case/cover; adapter cards

1. Place the chassis flat on the table. Position it so that you are looking at it from the front.

2. Locate and remove floppy drive mounting screws as shown in Figure 4-14. Place screws in container and label container "floppy drive mounting screws."

Fig. 4-14. Remove AT Floppy Drive Mounting Screws

Remove floppy drive mounting screws
(Note: screws sometimes are on bottom of drive)



3. Remove screws at front of drive, holding metal tabs and/or keeper bar as shown in Figure 4-15.

4. Slide the floppy drive forward about two inches, and then disconnect the power connector and signal connector at the top-rear of the drive (Figure 4-16).

5. Slide the floppy drive all the way out and set it aside.

6. Locate and remove the hard drive mounting screws as shown in Figure 4-17. Place screws in container and label as "hard drive mounting screws."

7. Slide the hard drive forward about two inches.

Fig. 4-15. Remove Floppy Drive Tab screws

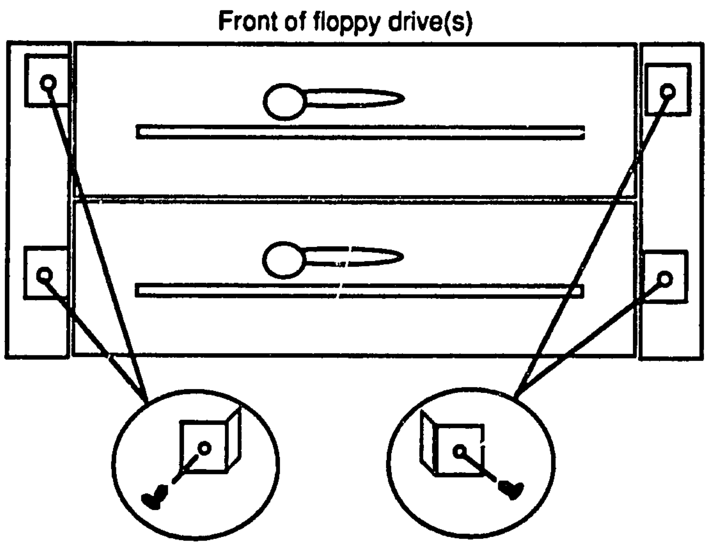
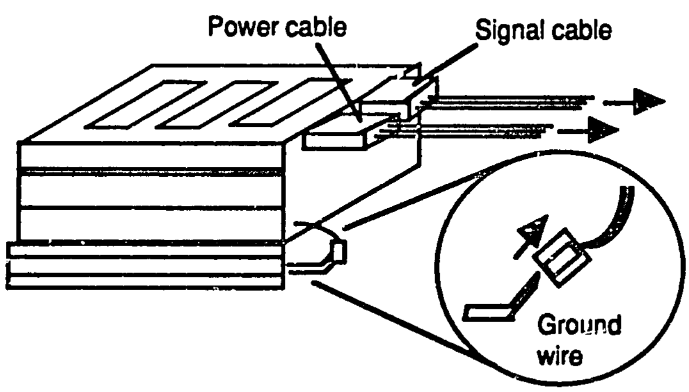


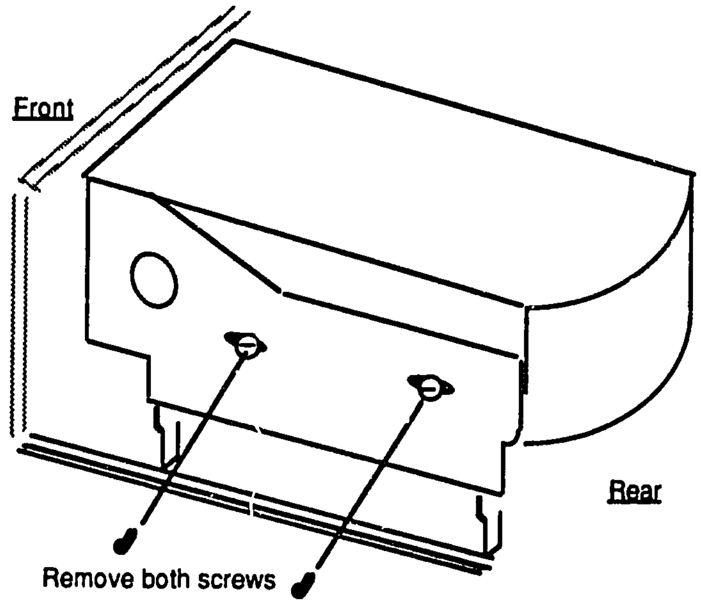
Fig. 4-16. AT Floppy Drive Cable Connections



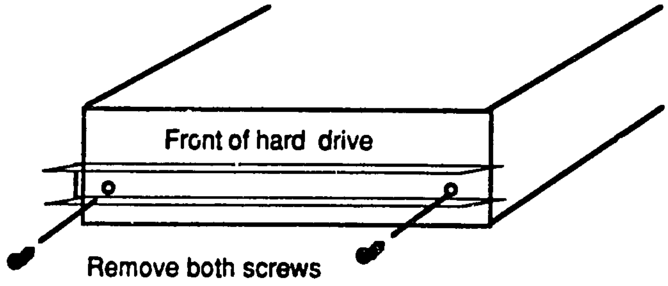
CAUTION: Do not pull on wires or cables. Grasp and pull on the connectors.

Fig. 4-17. Hard Drive Mounting Screws

(a) Mounting screws



(b) Keeper bar

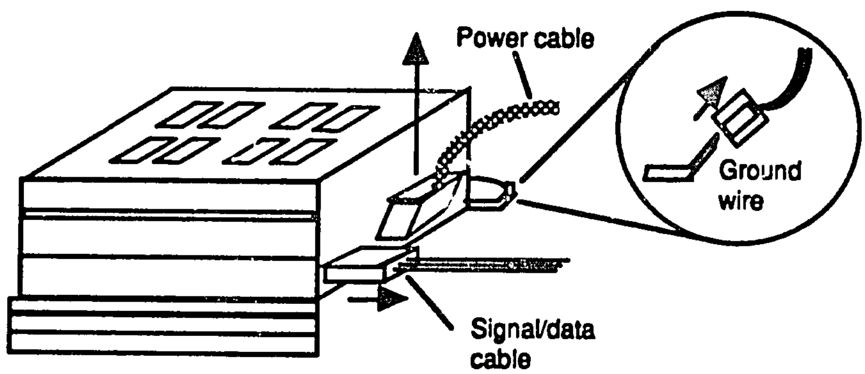


8. At the lower rear of the drive (Figure 4-18), label and disconnect:

- Data cable
- Control cable
- Power cable
- Ground lead

9. Slide the hard drive out and set it aside.

Fig. 4-18. AT Hard Drive Cable Connections



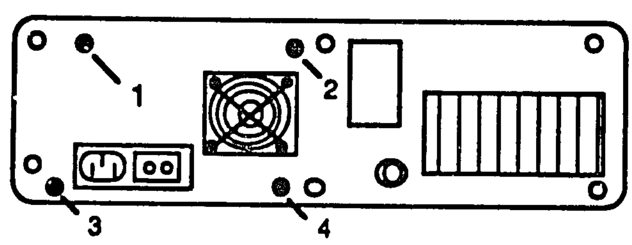
Remove Power Supply

Items already removed: cover; adapter cards; floppy drive; hard drive

Note: To remove power the supply, four mounting screws must be removed and the power supply must be slid forward to disengage the retaining tabs beneath it.

1. At the rear of the chassis, locate and remove the four (4) power supply retaining screws (Figure 4-19).

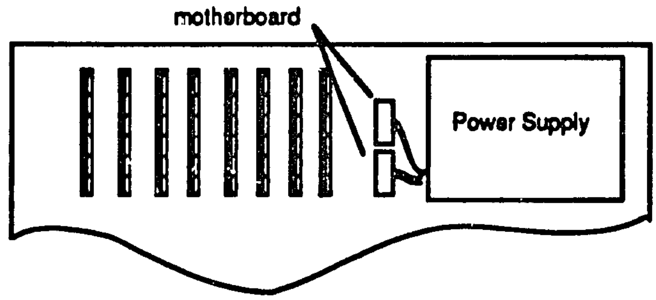
Fig. 4-19. AT Power Supply Retaining Screws



2. Label and disconnect the cables between the power supply and the motherboard (Figure 4-20).

CAUTION: Do not pull on wires or cables. Grasp and pull on the connectors.

Fig. 4-20. AT Power Supply Cable to Motherboard

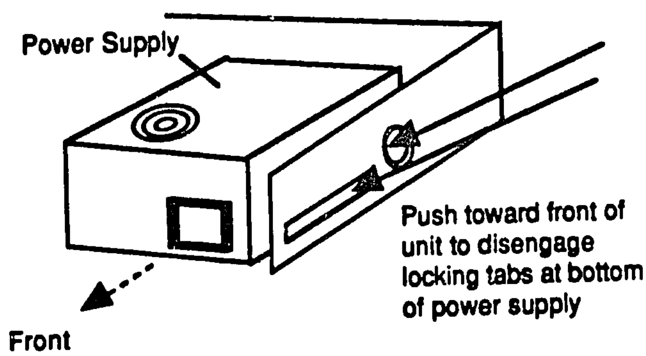


3. If not already disconnected, label and disconnect the cables between the power supply and the disk drives.

4. Slide power supply outward to disengage the retaining tabs beneath the supply (Figure 4-21).

5. Lift out the power supply and set it aside.

Fig. 4-21. Disengage AT Power Supply Retaining Tabs



Remove Motherboard

Items already removed: cover; adapter cards; floppy drive; hard drive; power supply

The motherboard is held in place by retaining screws and by plastic "spacers" or "standoffs. The standoffs hold the motherboard away from the metal chassis to prevent short circuits. The standoffs fit into slots. After removing the retaining screws, the board must be slid sideways to disengage the standoffs from their slots.

1. Make a sketch of the motherboard, showing the position of all cables attached to it.

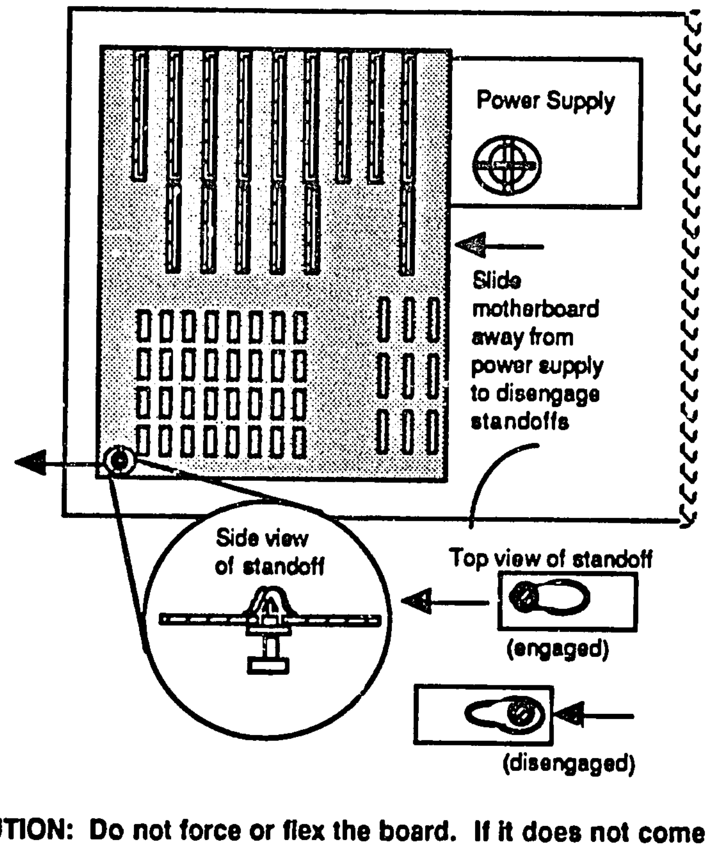
2. Label and disconnect all cables to the motherboard.

3. Remove the retaining screws; place screws in an container and label the container "motherboard retaining screws."

4. Disengage the standoffs by sliding the motherboard in the direction away from power supply location (Figure 4-22). Be sure that standoffs have been disengaged before going to next step.

5. Grasp the opposite edges of the motherboard and lift board straight up and out.

Fig. 4-22. AT Motherboard Standoffs



CAUTION: Do not force or flex the board. If it does not come out easily, check that you have removed all of the retaining screws and that the standoffs have been disengaged.

32

PS/2 Models 30 and 30-286 Disassembly

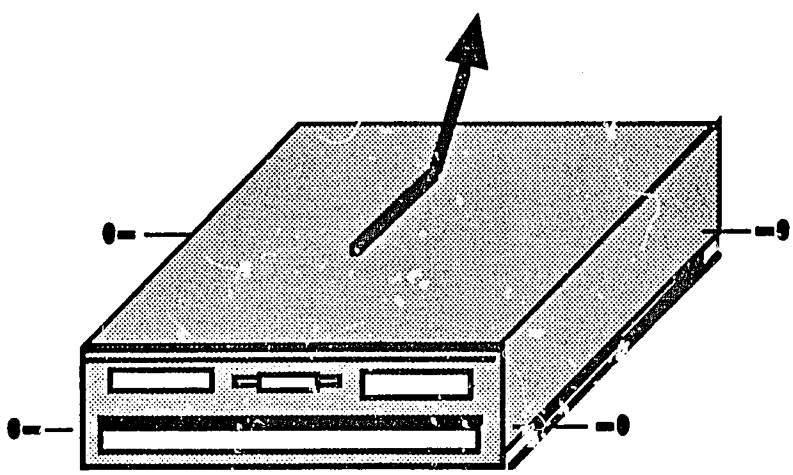
Components will be removed in the following order:

1. Cover
2. Rear Panel Cover
3. 3 1/2 Inch Floppy Drive
4. Hard Drive (Fixed Disk Drive)
5. Adapters
6. Bus Adapter
7. Power Supply
8. Single In-Line Memory Modules (SIMMS)
9. Motherboard (System board)

Remove System Cover

1. If keylock for cover is installed, unlock it.
2. Loosen four screws – two on the bottom corner of each side of the unit, as shown in Figure 4-23.
3. Slide cover toward back of unit; lift cover upward and off.

Fig. 4-23. Remove PS/2 Model 30 and 30-286 System Cover

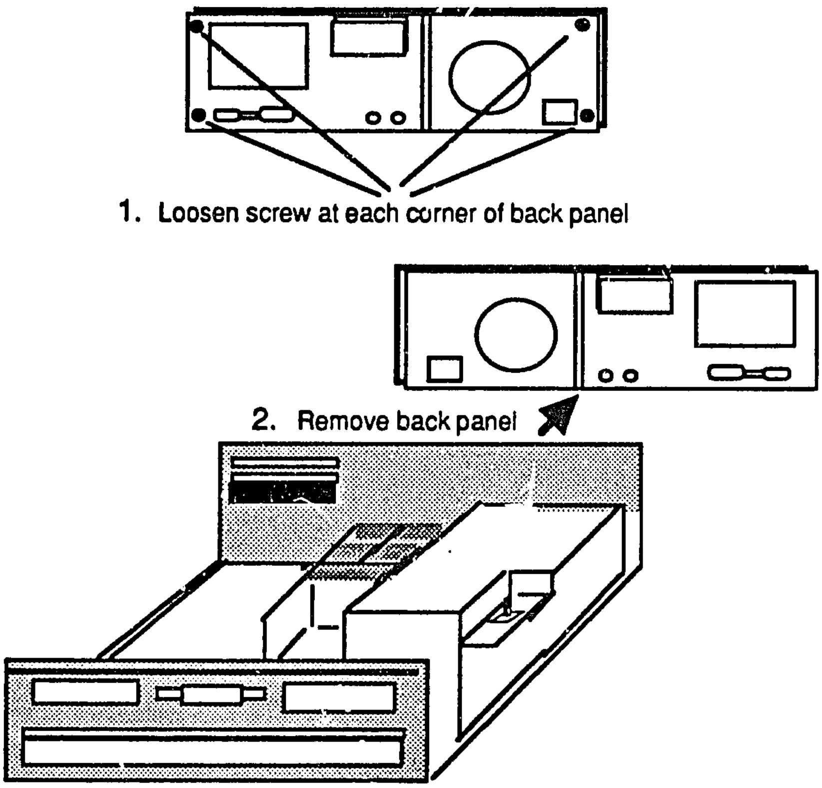


Remove Rear Panel Cover

1. Loosen four screws – one at each corner of the rear panel (Figure 4-24).

2. Pull cover off, straight away from back of unit.

Fig. 4-24. Remove Rear Panel Cover



Remove 3 1/2 Inch Floppy Drive

1 Insert shipping cardboard or a scratch disk into the drive.

2. At front of disk drive (Figure 4-25), remove plastic bezel (cover).

3. Press down on two tabs at top of bezel.

4. Pull bezel straight out, away from unit.

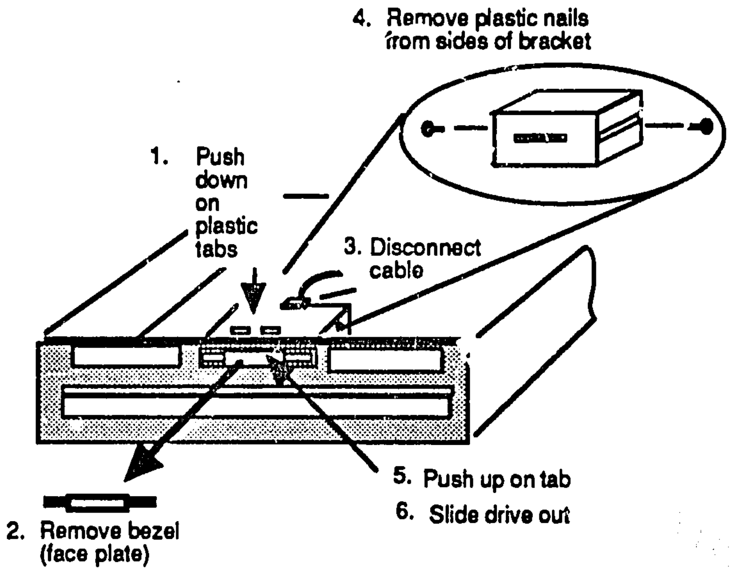
5. At the back of the drive, detach cable connector.

6. At each side of the drive, remove the plastic nail; two nails total.

7. Beneath front of drive, press up on tab to release drive.

8. Pull drive out; set it aside.

Fig. 4-25. Remove Bezel and Floppy Drive



Remove Hard Drive (Fixed Disk Drive)

The procedure is the same as for the 3 1/2 inch floppy drive as described above. (Refer to Figure 4-25.)

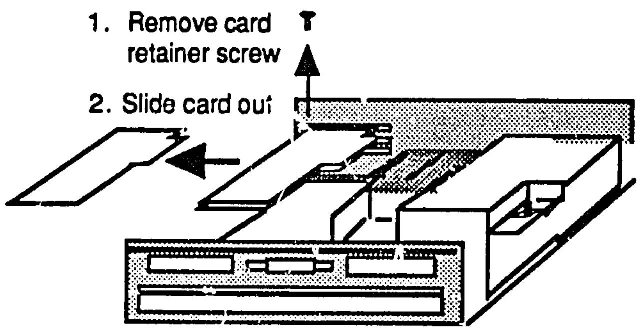
1. At front of disk drive, remove plastic bezel (cover).
2. Press down on two tabs at top of bezel.
3. Pull bezel straight out, away from unit.
4. At the back of the drive, detach cable connector.
5. At each side of the drive, remove the plastic nail; two nails total.
6. Beneath the front of the drive, lift up on tab to release the drive.
7. Pull the drive out; set it aside.

Remove Adapters

The adapters slide into the unit from the side. The cards actually plug into a "bus adapter" (to be removed after removing the cards). Each adapter card is held in place by a retainer screw at the left rear of the card (as you face the unit from the front). The screw attaches to the card retainer bracket at the rear of the unit.

1. Remove the retainer screw at left rear of card (Figure 4-26).
2. Grasp the card at both ends and slide it sideways, out of its slot.

Fig. 4-26. PS/2 Model 30 and 30-286 Adapter Removal



Remove Bus Adapter

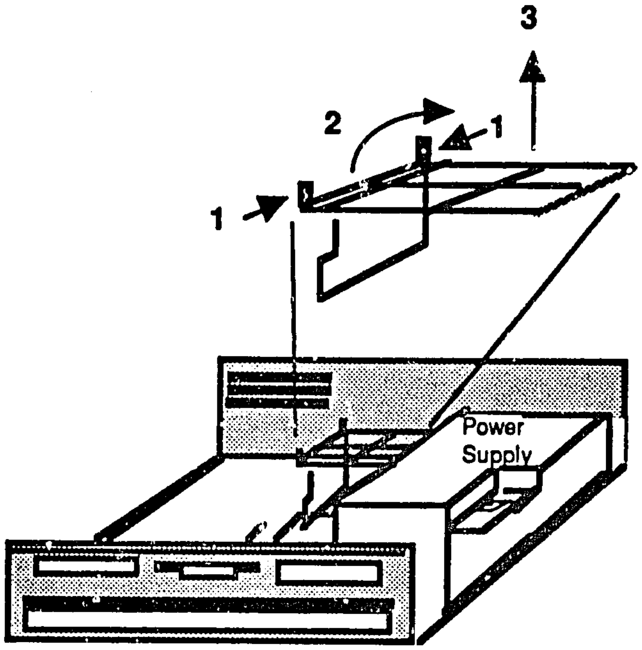
The bus adapter is held in place by two tabs that insert into the case of the power supply (Figure 4-27).

1. At top of bus adapter support, press inward on two tabs.

2. Grasp the support and rotate it upward to disengage the tabs that extend into the power supply.

3. Lift adapter support up and out.

Fig. 4-27. PS/2 Model 30 and 30-286 Bus Adapter Removal



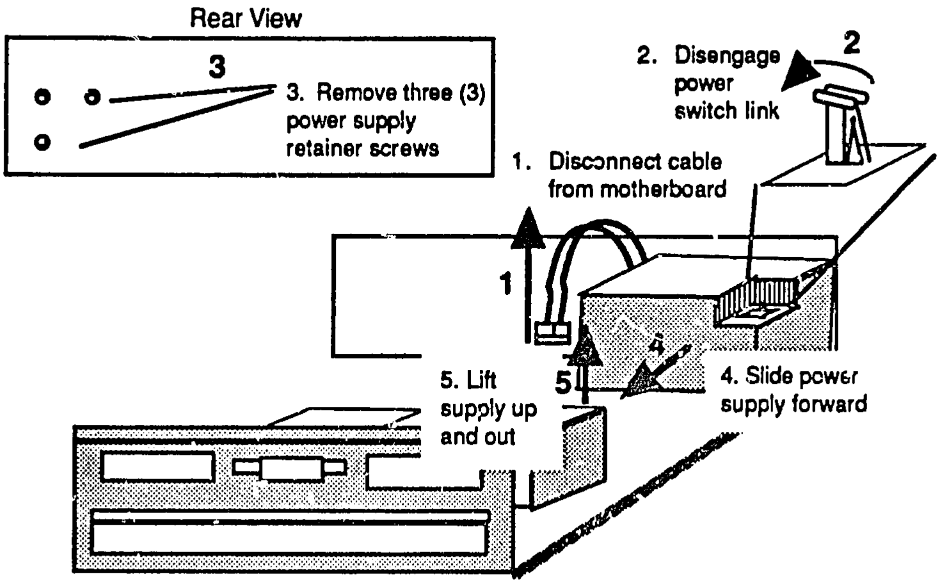
Remove Power Supply

The power supply cable is attached to the motherboard. The supply itself is held in place by three screws through the back panel, and by tabs underneath it that engage the bottom panel of the chassis. Also, the power supply switch is attached to a link that must be disengaged before removing the supply (Figure 4-28).

Note: To remove the power supply, the rear panel, adapter cards and bus adapter support must be removed first. The disk drives do not need to be removed.

1. Label and detach power supply power cable from the motherboard – grasp the connector and pull it straight up.
2. Disengage the power switch link – lift it upward and toward the front of the PC.
3. At the back of the power supply, remove the three retaining screws.
4. Slide supply toward front of PC to disengage tabs at bottom of supply.
5. Lift supply up and out.

Fig. 4-28. PS/2 Model 30 and 30-286 Power Supply Removal



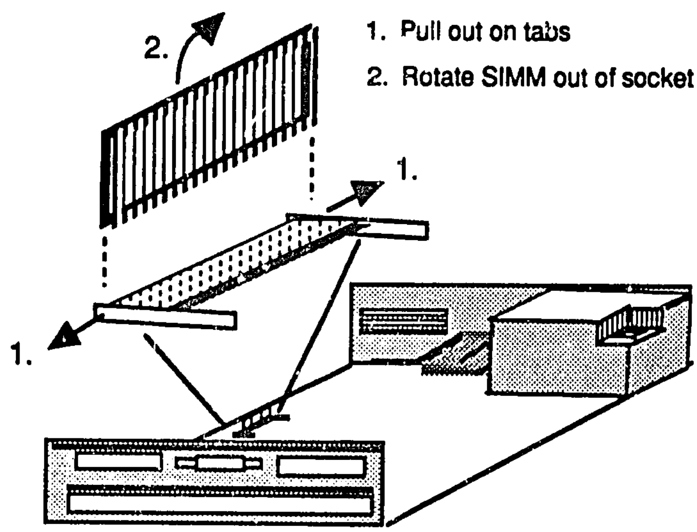
88

Remove Single In-Line Memory Modules (SIMMS)

Simms are held in place by two tabs, one at each end of the component (Figure 4-29).

1. Gently pull outward (away from the SIMM) on each tab at the same time.
2. Lift SIMM out of its socket. Gently rotating (rocking) the SIMM will help to ease it out. Take care not to damage the socket!

Fig. 4-29. PS/2 Model 30 and 30-286 SIMM Removal

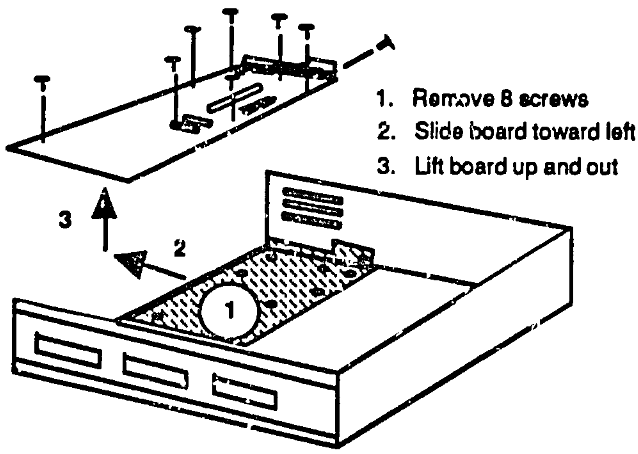


Remove Motherboard

Eight screws are used to hold the motherboard in place (Figure 4-30).

1. Remove all eight screws.
2. Slide the board sideways, away from the power supply location.
3. Lift the board up and out.

Fig. 4-30. PS/2 Model 30 and 30-286 Motherboard Removal



PS/2 Models 50, 50Z and 70 Disassembly

Components will be removed in the following order:

1. Cover
2. Battery and Speaker Assembly
3. Fan Assembly
4. Adapters
5. 3 1/2 Inch Floppy Drive
6. Hard (Fixed) Disk Drive and Adapter Card
7. Drive Support Structure
8. Power Supply
9. Motherboard (System Board)
10. Single In-line Memory Modules (SIMMS)

Remove Cover

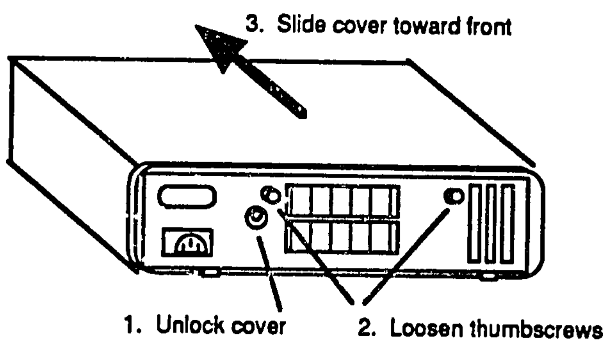
The cover is held in place by two thumbscrews near the top of the rear panel (Figure 4-31). A screwdriver is not needed. Also, the cover keylock must be unlocked before the cover can be removed.

1. Unlock the cover keylock.

2. At the rear panel, near the top, loosen two thumbscrews.

3. Slide cover toward front of PC, then lift cover up and off.

Fig. 4-31. Remove Model 50/70 Cover



Remove Battery and Speaker Assembly

The battery and speaker assembly is located at the left front corner of the PC. The battery and speaker are assembled into a single unit, but the battery may be removed by itself. The battery should be removed first, to avoid accidentally shorting it out (discharging it) as you work.

Remove Battery

1. Release battery by pressing its retainer tabs toward the back of the PC.
2. Grasp battery and lift it straight up and out.

Remove Speaker Assembly

1. At bottom of assembly, press tab to release it.
2. Grasp assembly and lift it straight up and out. (Take care not to damage the speaker cone.)

Remove Fan Assembly

The assembly is held in place by two push-button fasteners, one at the top of each side of the assembly (Figure 4-32).

1. Obtain the pry tool that is stored at the right front corner of the PC (or use a thin blade screwdriver).
2. Using the tool, pry the two tabs upward to release the assembly.
3. Lift the assembly straight up and out.

Remove Adapters

1. Label adapters and corresponding slots (Figure 4-33).
2. Diagram adapter and corresponding slot positions.
3. Label and disconnect adapter cable(s).
4. At lower-rear corner of PC, loosen thumbscrew for each adapter card in place.
5. Grasp adapter at both ends and lift it up and out.

Fig. 4-32. Remove Model 50/70 Fan Assembly

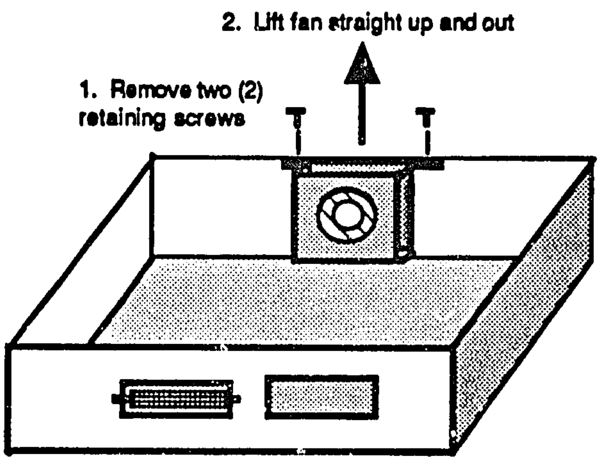
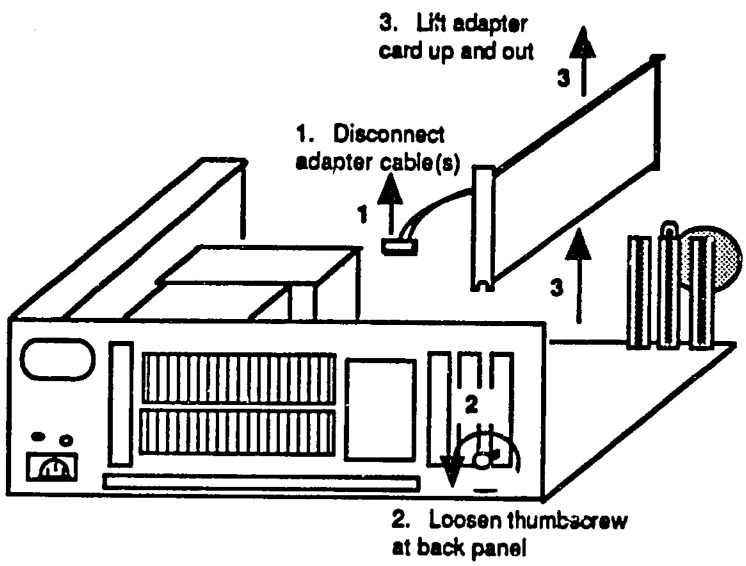


Fig. 4-33. Remove Model 50/70 Adapters



93

Remove 3 1/2 Inch Floppy Drive

The drive is held in place by just a single tab at the bottom front of the drive.

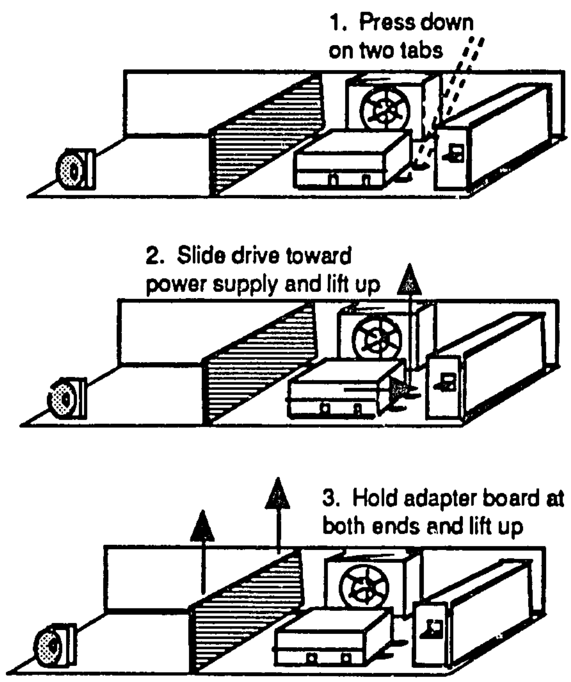
1. Press upward on retaining tab.
2. Slide drive out, away from PC.

Remove Hard (Fixed) Disk Drive and Drive Adapter Card

The drive is held in place by two tabs located at the base of the drive, next to the power supply (Figure 4-34).

1. Press down on the two tabs to release the drive.
2. Lift the drive straight up and out.
3. Grasp the adapter at both ends and lift it straight up and out.

Fig. 4-34. Remove Model 50/70 Hard Drive and Hard Drive Adapter



.94

Remove Drive Support Structure

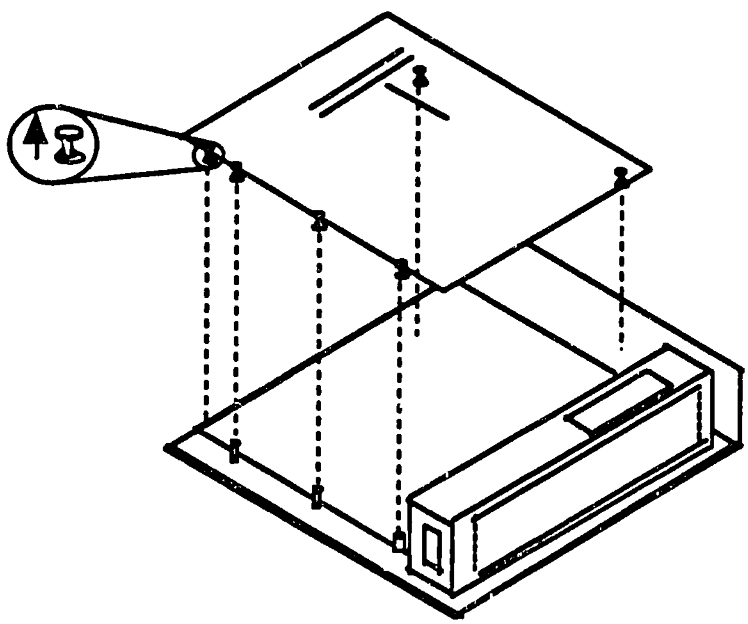
The structure is held in place by six, push-button fasteners (Figure 4-35). Four of the fasteners are located at the base of the structure, toward the front of the PC. The remaining two are located toward the rear.

1. Obtain the pry tool that is stored at the right front corner of the PC (or use a thin blade screwdriver).

2. Pry up the six push-button fasteners.

3. Lift the structure straight up and out.

Fig. 4-35. Remove Model 50/70 Drive Support Structure

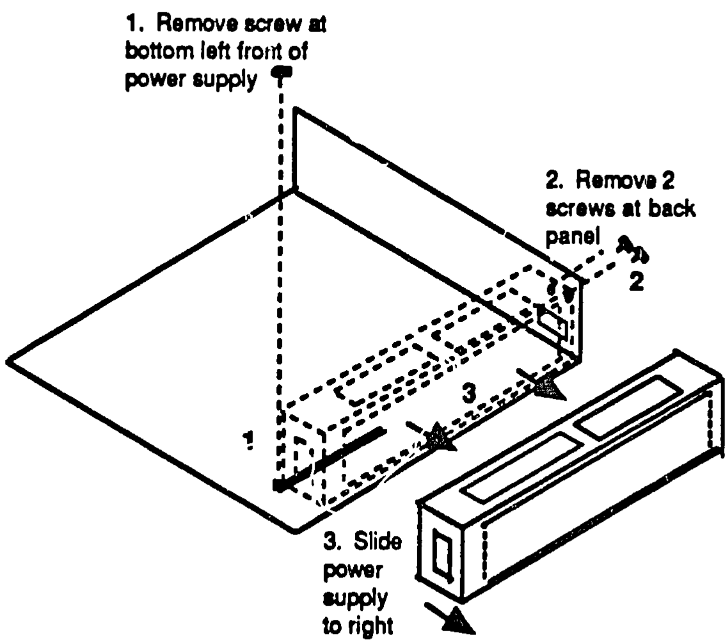


Remove Power Supply

The power supply is held in place by three screws (Figure 4-36). Once screw is at the front of the supply; the remaining two are at the back panel of the PC.

1. Remove screw at front of power supply.
2. Remove two screws at back of power supply (at rear panel of PC).
3. Slide the power supply sideways, out of the PC.

Fig. 4-36. Remove Model 50/70 Power Supply

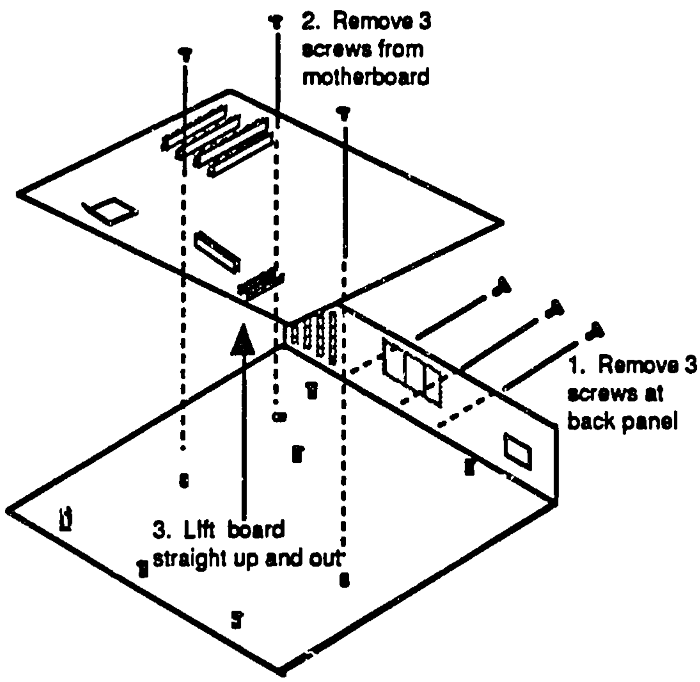


Remove Motherboard (System Board)

The motherboard is held in place by six screws (Figure 4-37). Three screws are at the back panel of the PC. The remaining three screws are on the board, pointing downward into the bottom panel of the chassis.

1. Remove three screws at the back panel of the PC.
2. Remove the three screws that hold the motherboard to the bottom panel of the chassis.
3. Lift the board straight up and out.

Fig. 4-37. Remove Model 50/70 System Board

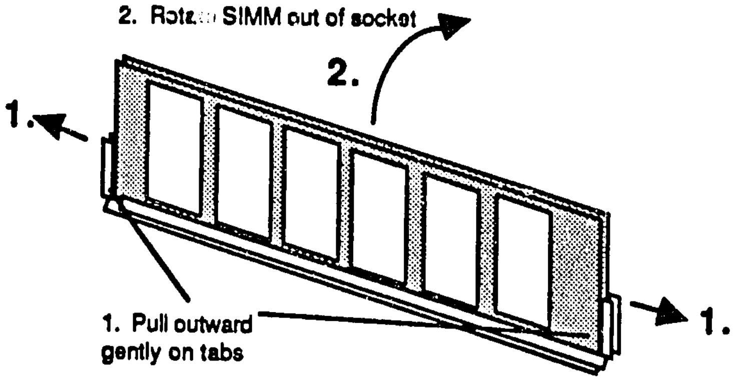


Remove Single In-line Memory Modules (SIMMS)

Simms are held in place by two tabs, one at each end of the component (Figure 4-38).

1. Gently pull outward (away from the SIMM) on each tab at the same time.
2. Lift SIMM out of its socket. Gently rotating the SIMM will help to ease it out. Take care not to damage the socket!

Fig. 4-38. Remove Model 50/70 SIMMS



98

PS/2 Models 60 and 80 Disassembly

Components will be removed in the following order:

1. Cover
2. Adapters
3. Battery and Speaker Assembly
4. Front Bezel
5. Power Supply
6. Floppy Drives
7. Floppy Drive Cable Retainer
8. Hard (Fixed) Drive D (If Present)
9. Hard (Fixed) Drive C
10. Hard (Fixed) Drive Support Structure
11. Motherboard (System board)

Remove Cover

The cover is held in place by a keylock and two thumbscrews (Figure 4-39).

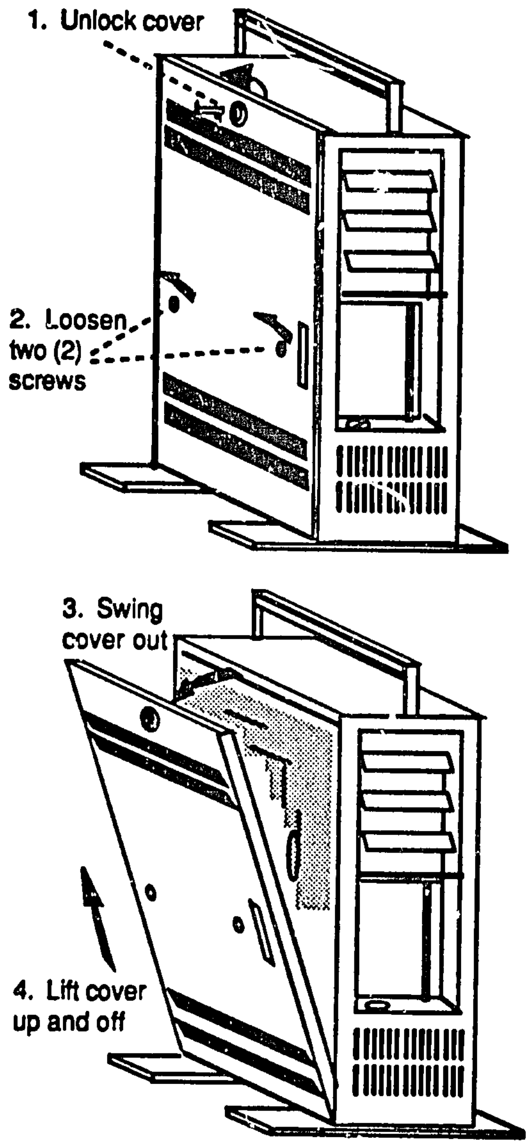
1. Unlock cover keylock.

2. At the side of the PC, loosen two thumbscrews.

3. Tilt the cover outward, away from the PC.

4. Lift cover upward and off.

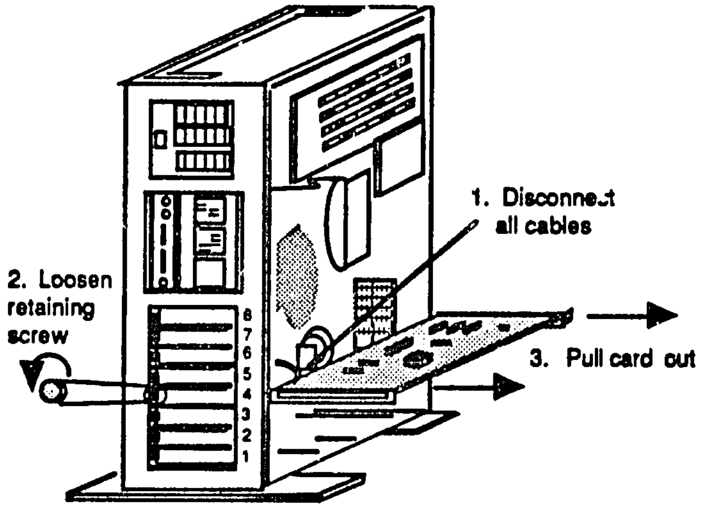
Fig. 4-39. Remove PS/2 Model 60/80 Cover



Remove Adapters

1. Label adapters and corresponding slots.
2. Diagram adapters and corresponding slot positions.
3. Label and detach all cables connected to the adapters (Figure 4-40).
4. Loosen thumbscrews that hold adapter in place.
5. Grasp the adapter at each end of its outer edge, and pull straight out. (Avoid handling or touching the adapter edge connector.)

Fig. 4-40. Remove PS/2 Model 60/80 Adapters

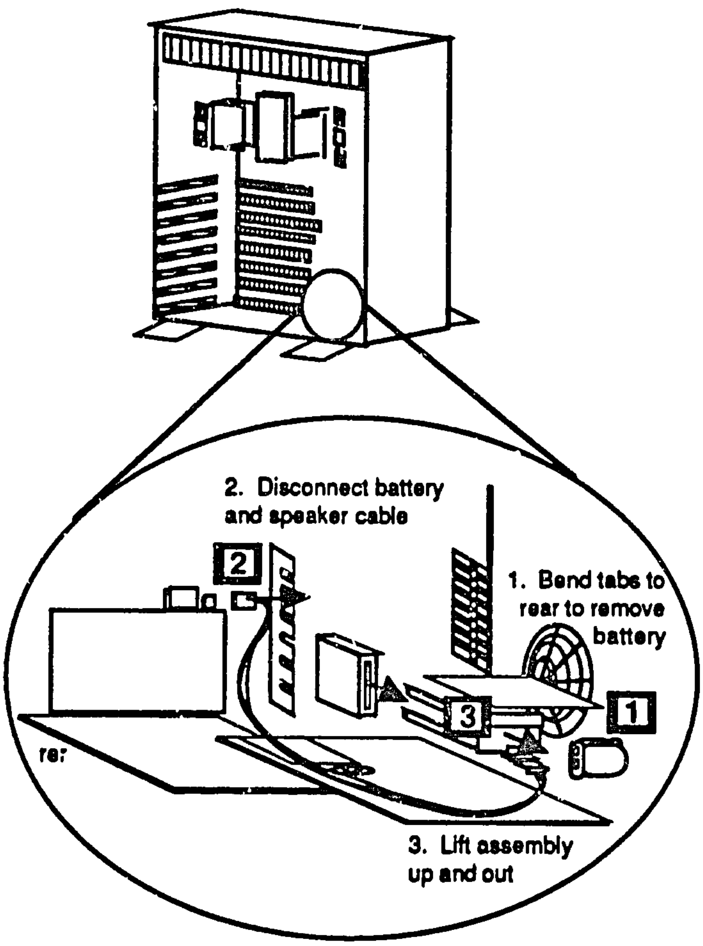


Remove Battery and Speaker Assembly

As you face the left side of the PC, the assembly is located in the lower, right hand corner (Figure 4-41). The battery and speaker are assembled into a single unit, but the battery may be removed by itself. The battery should be removed first, to avoid accidentally shorting it out (discharging it) as you work.

1. Press the battery-holder tabs to release battery.
2. Lift the battery up and out.
3. Detach the battery/speaker cable.
4. At the bottom of the assembly, press the tab to release the assembly.
5. Pull the assembly out.

Fig. 4-41. Remove PS/2 Model 60/80 Battery and Speaker Assembly



Remove Front Bezel

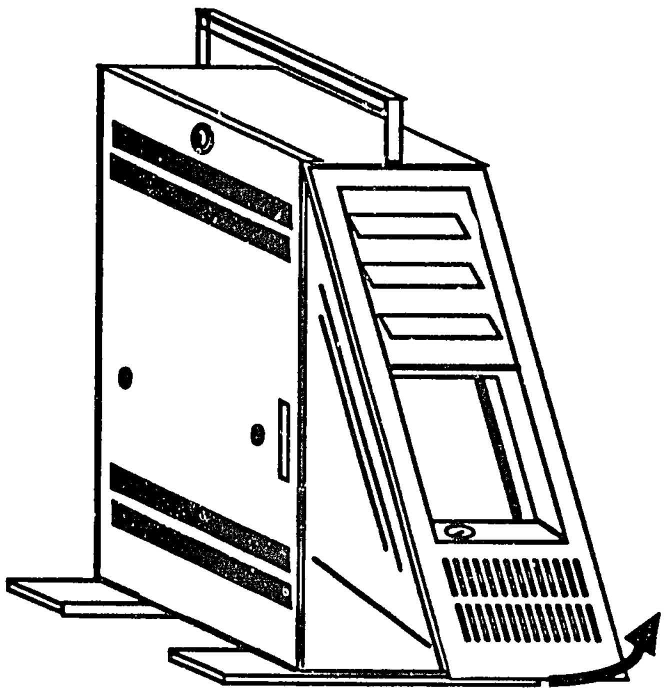
The bezel is a plastic frame (cover; panel) that protects the front of the PC. It must be removed in order to remove the power supply and the floppy drives. The bezel is not held by screws. It snaps on and off (Figure 4-42).

1. Position yourself in front of the PC.

2. Place your fingertips under the bottom edge of the bezel, or grasp the lower corners of the bezel.

3. Pull the bottom edge of the bezel toward you, and off.

Fig. 4-42. Remove PS/2 Model 60/80 Front Bezel

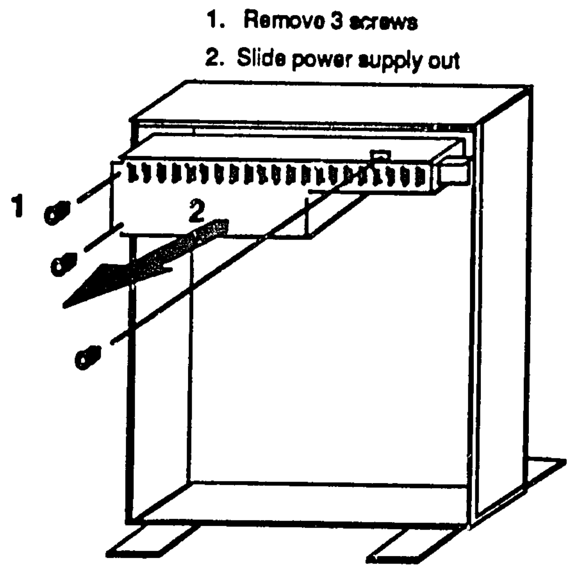


Remove Power Supply

The power supply connects to the unit with cables, and it is held in place by three screws (Figure 4-43). One screw is located toward the upper front of the supply, near the power switch. The remaining two screws are at the back. As you face the side of the PC, the two screws are to your left, at the end of the supply.

1. Disconnect all cables from the power supply.
2. Remove the three retaining screws.
3. Carefully support the supply from the bottom (to avoid dropping it), and slide it out.

Fig. 4-43. Remove PS/2 Model 60/80 Power Supply



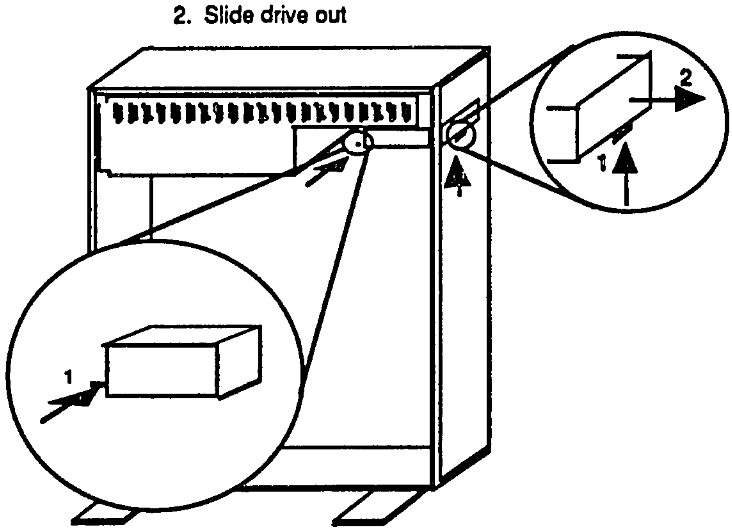
Remove Floppy Drive

The drive is held in place by two catches (Figure 4-44). One (usually colored black) is at the back of the drive. The other is at the lower front of the drive.

1. Press upward on the tab located at the rear of the drive. It should "click" when it is released.
2. Press upward on the tab at the front of the drive.
3. Slide the drive out.

Fig. 4-44. Remove PS/2 Model 60/80 Floppy Drive

1. Press up on tab at front, and press inward on tab at back at the same time
2. Slide drive out

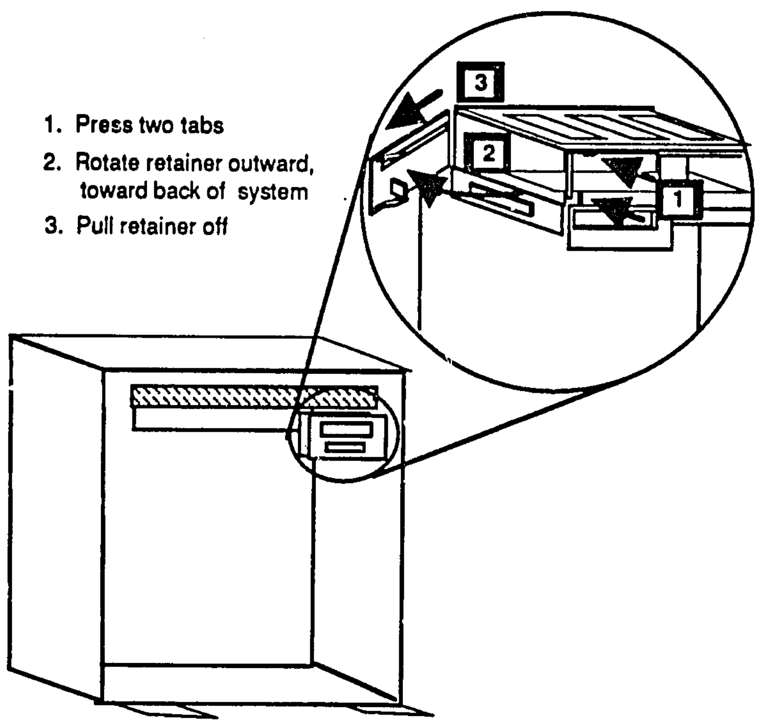


Remove Floppy Drive Cable Retainer

When the floppy drives are plugged in, their cables are held in place by a retainer (Figure 4-45). The cable retainer itself is held in place by two tabs.

1. At the sides of the retainer, press the two tabs.
2. Grasp the retainer and rotate it toward the back of the PC.
3. Pull the retainer off.

Fig. 4-45. Remove PS/2 Model 60/80 Floppy Drive Cable Retainer



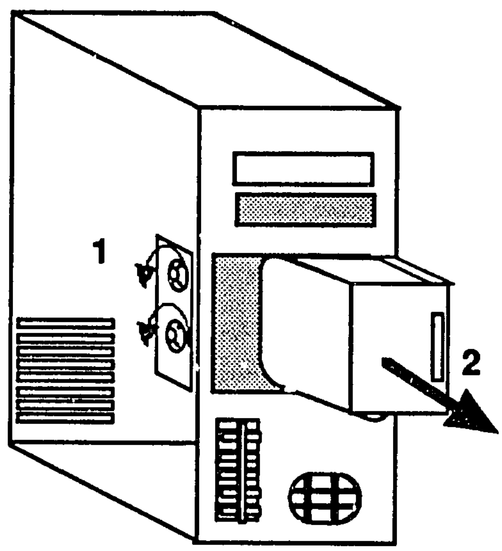
Remove Hard (Fixed) Drives D and C

Because of the way the drives are installed, drive D (if present) must be removed before removing drive C. The drives are connected to the system by cables and a ground wire. Each drive is held in place by two thumbscrews (Figure 4-46).

1. Disconnect all cables and ground wire.
2. Loosen two thumbscrews (turn counterclockwise).
3. Slide drive out.

Fig. 4-46. Remove PS/2 Model 60/80 Fixed Drive D

1. Loosen 2 thumbscrews (counter-clockwise)
2. Slide drive out



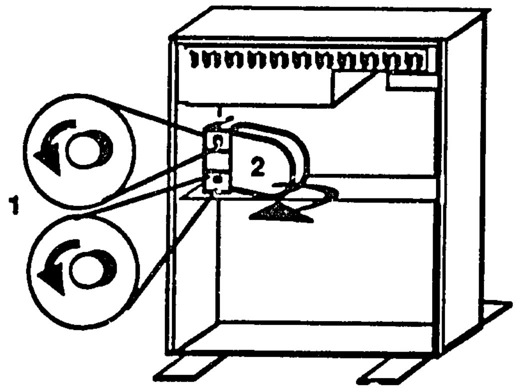
Remove Hard (Fixed) Drive C

Drive removal is depicted in Figure 4-47.

1. Label and disconnect all cables and ground wire.
2. Loosen two thumbscrews (turn counterclockwise).
3. Slide drive slightly toward front of PC.
4. Lift drive out.

Fig. 4-47. Remove PS/2 Model 60/80 Fixed Drive C

1. Loosen 2 thumbscrews (counter clockwise)
2. Slide drive toward front and lift out



Remove Hard (Fixed) Drive Support Structure

A metal structure ("cradle") is used to support the hard drives. This structure must be removed in order to remove the motherboard. The structure is held in place by four screws (Figure 4-48).

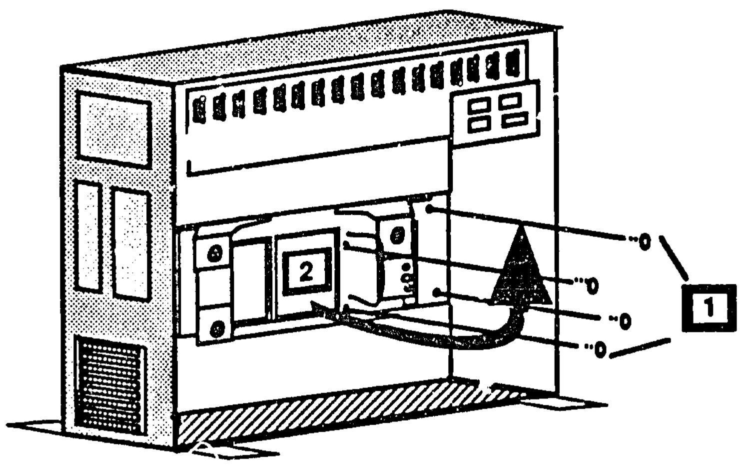
1. Remove the four retaining screws.

2. Slide the structure toward the front of the PC.

3. Lift the structure up and out.

Fig. 4-48. Remove PS/2 Model 60/80 Drive Support Structure

1. Remove 4 screws
2. Slide structure forward, lift out



Remove Motherboard (System Board)

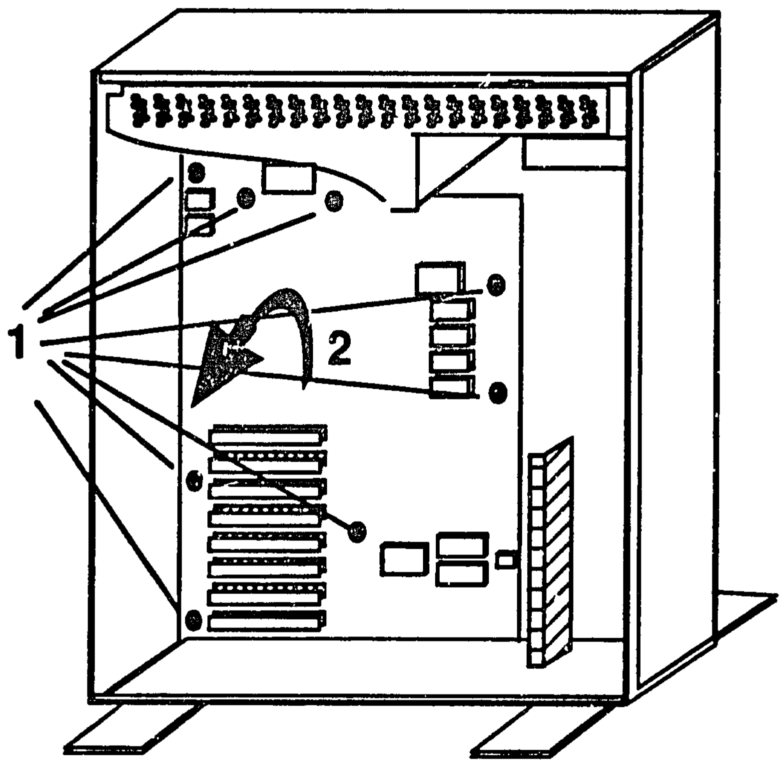
The motherboard is connected tc the unit by cables. The board is held in place by eight retaining screws (Figure 4-49).

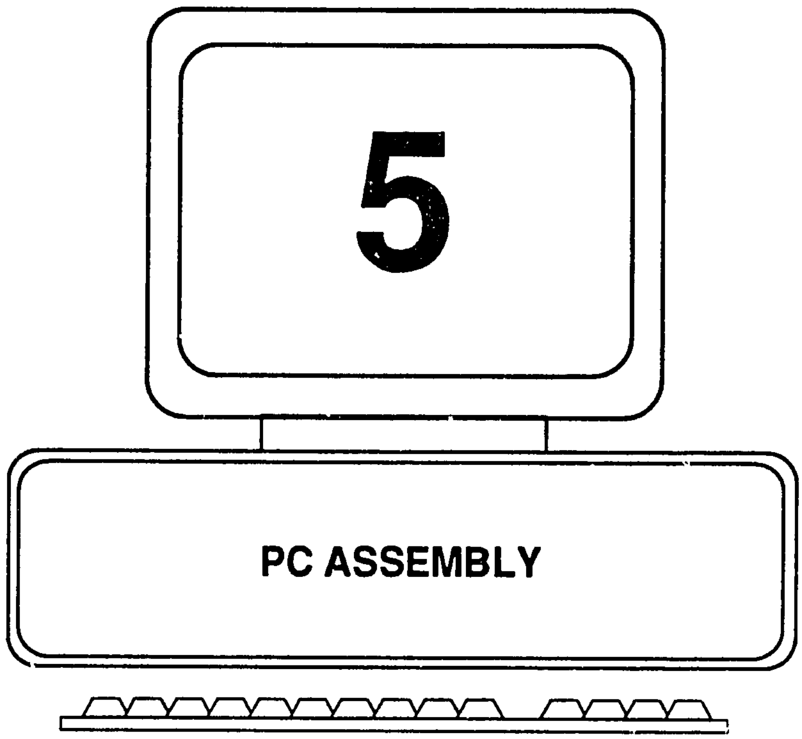
1. Remove the eight retaining screws.

2. Lift the board up and out.

Fig. 4-49. Remove PS/2 Model 60/80 Motherboard

1. Remove 8 retaining screws (●)
2. Lift board up and out




5
PC ASSEMBLY

CHAPTER 5: PC ASSEMBLY

PC, XT, AT AND PS/2 MODELS

General Guidelines

Before you begin, be sure that you:

- Discharge static electricity from your body by touching a metal doorknob or a piece of metal furniture.
- Allow yourself enough room in which to work. Preferably, move the PC to an uncluttered area, where you can lay out the components in order, before installing them.
- Have available the notes, sketches and diagrams that you made during disassembly. Use them. Do not trust to memory or trial and error. Doing so will create problems and may damage the system.
- Plan to work slowly, carefully and methodically. In the long run, you will accomplish more in less time than if you rush. And, you will avoid errors that could make matters worse than they were before you started.

PC and XT Assembly

Components will be installed in the following order:

1. Motherboard
2. Power Supply
3. Disk Drives
4. Adapter Boards
5. Case/Cover

Work slowly, carefully and gently. Never force, twist or bend components. Components will be installed easily if proper procedures are followed.

Install Motherboard

Items already installed: none

The motherboard is held in place by retaining screws and by plastic "spacers" or "standoffs." The standoffs hold the motherboard away from the metal chassis to prevent short circuits. Be sure the board is seated properly on the standoffs before installing the retaining screws. Otherwise, the board may bend and become damaged (Figure 5-1).

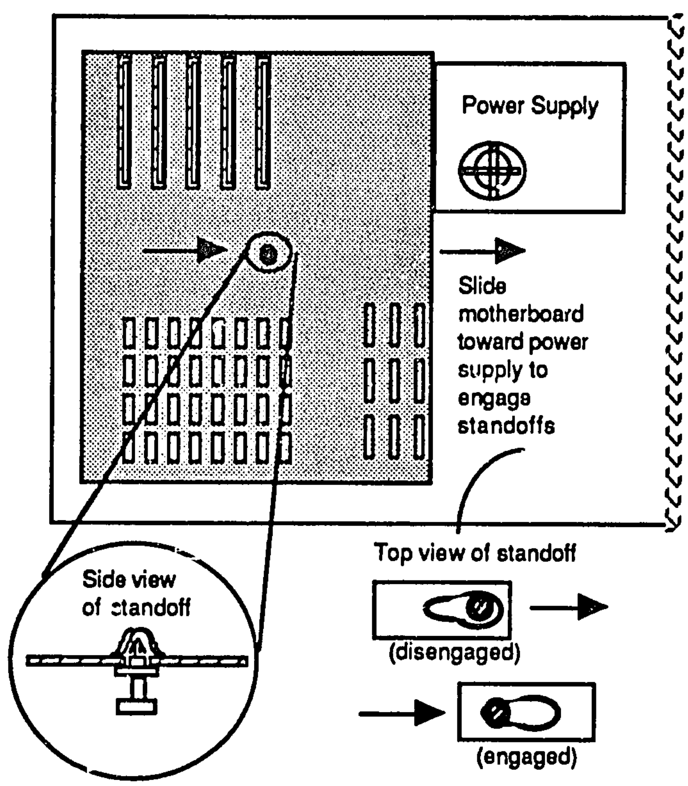
1. Grasp opposite edges of motherboard and lower it straight into the unit. Align the board so the standoffs protrude through their openings in the board.

   CAUTION: Do not force or flex the board. If it does not install easily, check that there is nothing under it (such as a stray screw).

2. Engage the standoffs by sliding the motherboard in the direction toward the location of the power supply; be sure every standoff is properly seated before going to next step.

3. Install the retaining screws.

Fig. 5-1. Engage Motherboard Standoffs



115

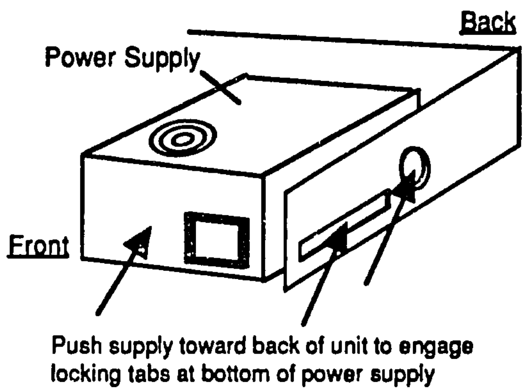
Install Power Supply

Items already installed: motherboard

To install power supply, the power supply must be slid toward the rear of the unit, to engage retaining tabs beneath it. Then the four mounting screws are installed.

1. Lower the supply into the unit and slide it toward the rear to engage the tabs beneath it (Figure 5-2).

Fig. 5-2. Engage Power Supply Retaining Tabs



2. Connect the cables between the power supply and the motherboard (Figure 5-3).

3. At the rear of the chassis, install the four (4) power supply retaining screws (Figure 5-4).

116

Fig. 5-3. Connect Power Supply Cable to Motherboard

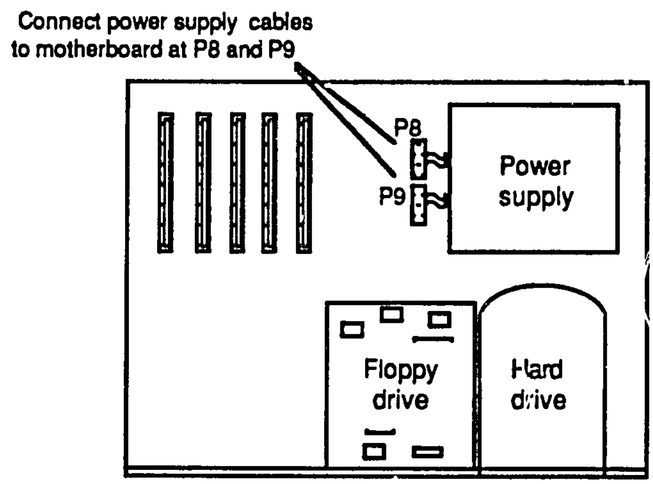
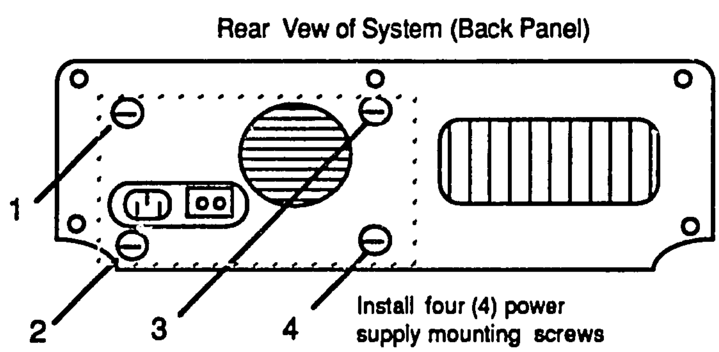


Fig. 5-4. Install Power Supply Retaining Screws



Install Disk Drives

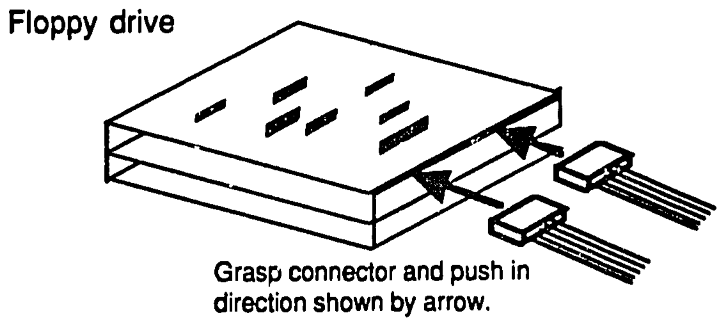
Items already installed: motherboard; power supply

Install Floppy Drive

1. Place the floppy drive into the unit.

2. Slide the drive about two inches toward the rear of the unit, and then connect the power connector and signal connector at the top-rear of the drive (Figure 5-5).

Fig. 5-5. Connect Floppy Drive Cables



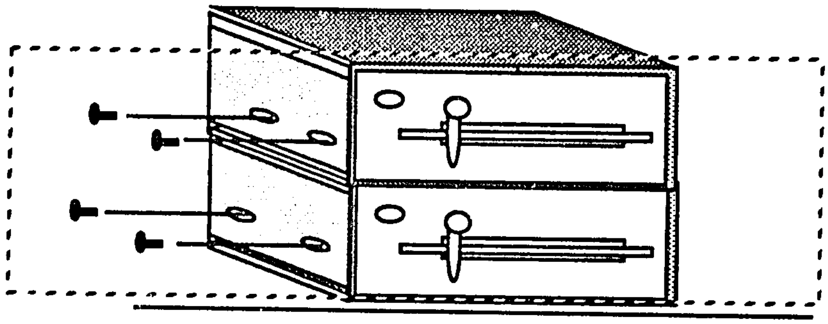
115

3. Install floppy drive mounting screws (Figure 5-6).

4. Remove the scratch disk from the drive.

Fig. 5-6. Install Floppy Drive Mounting Screws

Install floppy drive mounting screws
(Note: screws sometimes are on bottom of drive)



Install Hard Drive

1. Place the hard drive into the unit.

2. Slide the drive forward about two inches toward the back of the unit.

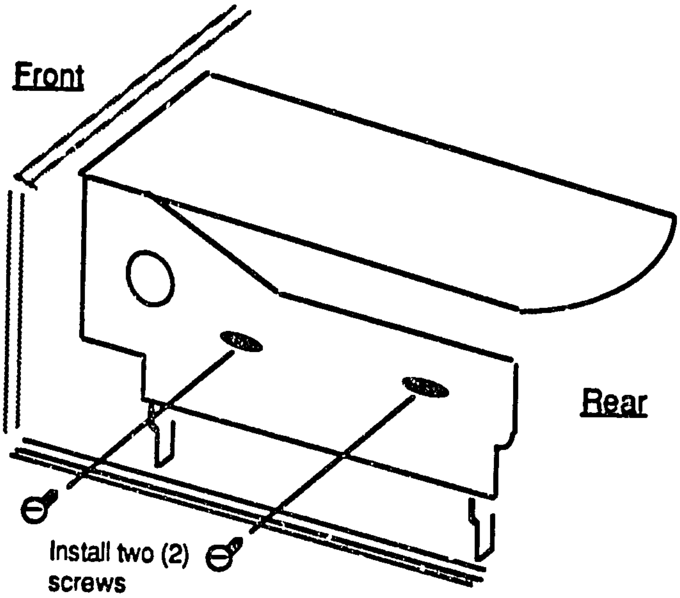
3. At the lower rear of the drive, connect:

   - Data Connector
   - Control Connector
   - Power Connector

4. Install the hard drive mounting screws (Figure 5-7).

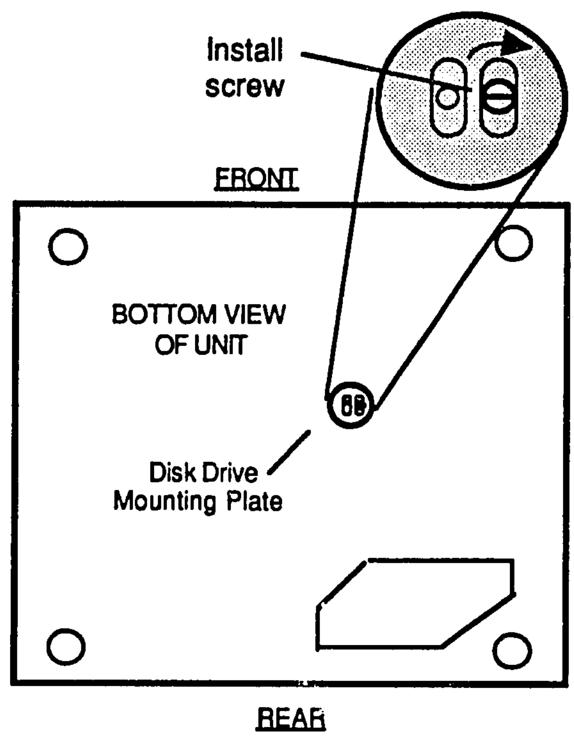
119.

Fig. 5-7. Install Hard Drive Mounting Screws



5. Stand the unit on end, with the front panel facing upward.
6. Position unit so that you are looking at the bottom of the chassis.
7. Install the hard drive retainer screw at the bottom of the chassis (Figure 5-8).

Fig. 5-8. Install Hard Drive Retainer Screw



Install Adapter Boards/Cards

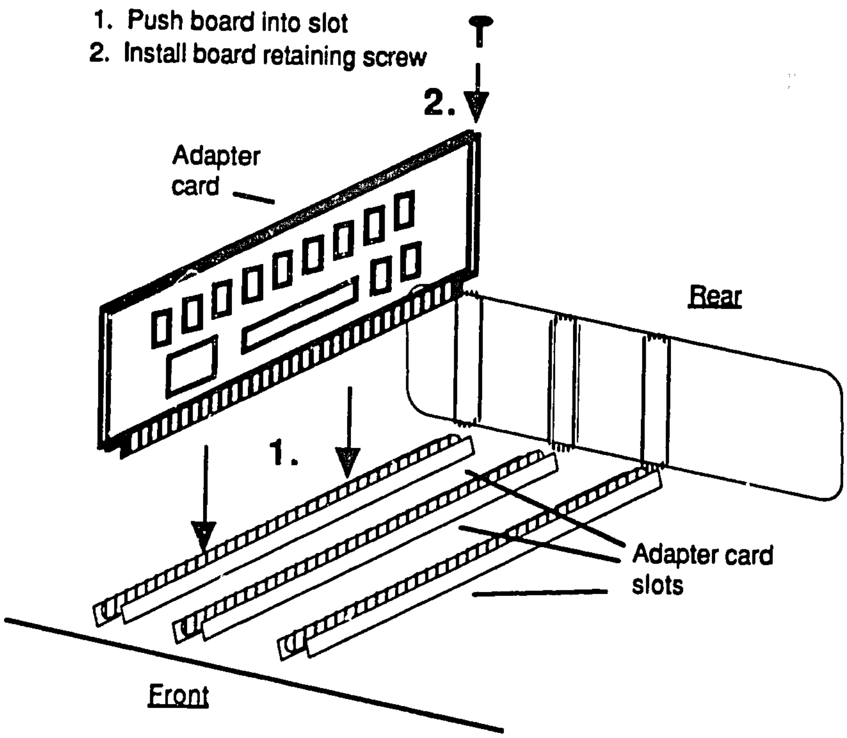
Items already installed: motherboard; power supply; disk drives

Refer to your sketch of the adapters made during disassembly. Odd-colored stripe on ribbon connector indicates pin number one (Pin #1). On the board, switches are labeled "SW" and jumpers are labeled "J."

1. Grasp adapter at both ends and align edge connecter with adaptor slot.
2. Press adapter downward, evenly, into the slot; do not twist or bend adapter.
3. Install the retainer screw that hold the adapter in place; there is one retainer screw for each adapter (Figure 5-9).
4. Connect adapter cables in accordance with the drawings you made during disassembly.

-121

Fig. 5-9. Install Adapter Board/Card



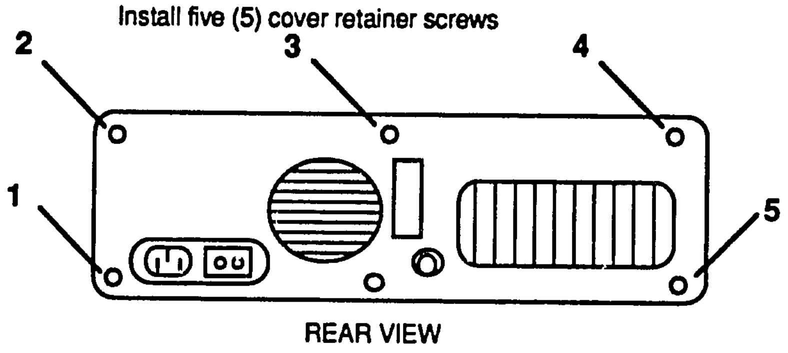
Install Cover

1. Place system unit on table or workbench with adequate room to work; position so you are looking at the front panel.

2. Align the rear of the cover with the front of the unit and slide the cover onto the unit; take care not to snag any interior cables; also see that the lower edges of the cover are properly aligned when sliding the cover on.

3. At rear panel, install five (5) screws (Figure 5-10).

Fig. 5-10. Install Cover Screws



AT Assembly

The procedures here are essentially the same as for the PC/XT models. For the AT, however, the disk drives are mounted on plastic rails. The drives slide in and out on these rails. Also, the drives are held in place by brackets and/or a "keeper bar."

Components will be installed in the following order:

1. Motherboard
2. Power Supply
3. Disk Drives
4. Adapter Boards
5. Cover

Work slowly, carefully and gently. Never force, twist or bend components. Components will be installed easily if proper procedures are followed.

Install Motherboard

Items already installed: none

The motherboard is held in place by retaining screws and by plastic "spacers" or "standoffs." The standoffs hold the motherboard away from the metal chassis to prevent short circuits. Be sure the board is seated properly on the standoffs before installing retaining screws. Otherwise, the board may bend and become damaged.

1. Grasp the opposite edges of motherboard and lower it straight into the unit.

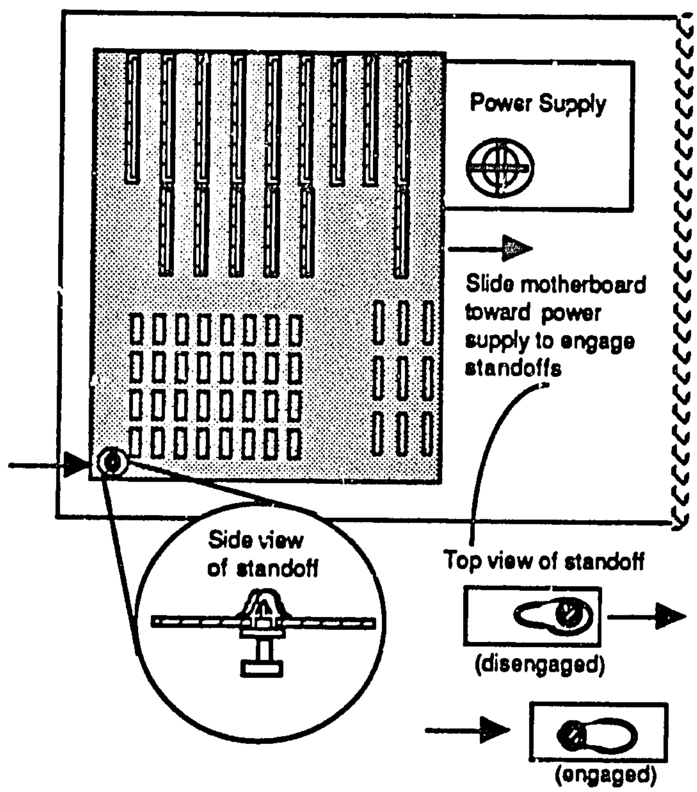
2. Align the board so standoffs come through their slots in the board.

   CAUTION: Do not force or flex the board. If it does not install easily, check that there is nothing under it (such as a stray screw).

3. Engage the standoffs by sliding the motherboard in the direction toward the location of the power supply (Figure 5-11); be sure every standoff is properly seated before going to next step.

4. Install the motherboard retaining screws.

Fig. 5-11. Engage Motherboard Standoffs



Install Power Supply

Items already installed: motherboard

To install power supply, the power supply must be slid toward the rear of the unit, to engage retaining tabs beneath it. Then the four mounting screws are installed.

1. Lower the supply into the unit and slide it toward the rear to engage the tabs beneath it (Figure 5-12).

2. Connect the cables between the power supply and the motherboard (Figure 5-13).

3. At the rear of the chassis, install the four (4) power supply retaining screws (Figure 5-14).

Fig. 5-12. Engage Power Supply Retaining Tabs

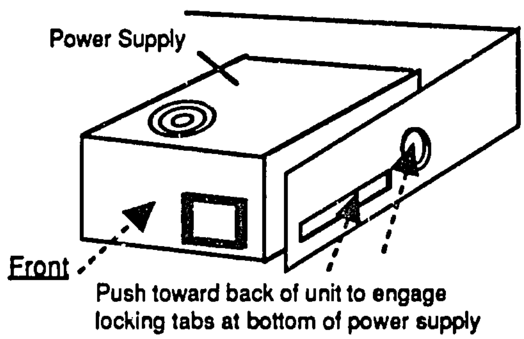


Fig. 5-13. Connect Cable from Power Supply to Motherboard

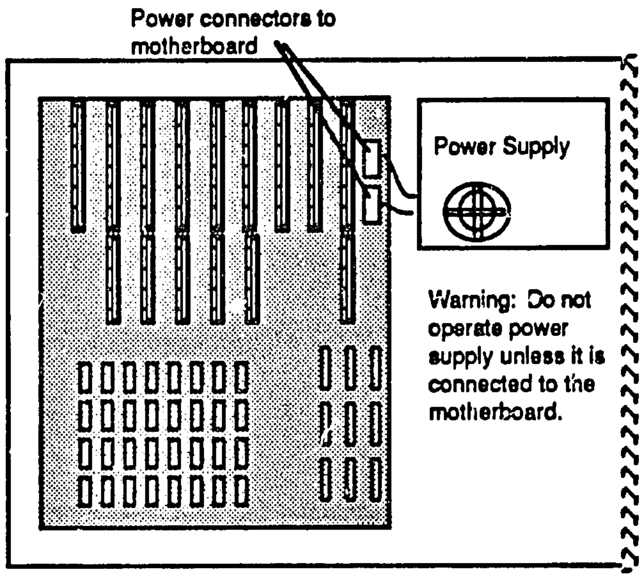
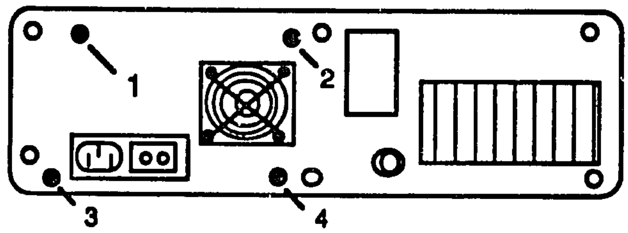


Fig. 5-14. Install Power Supply Retainer Screws

Install four (4) power supply retainer screws



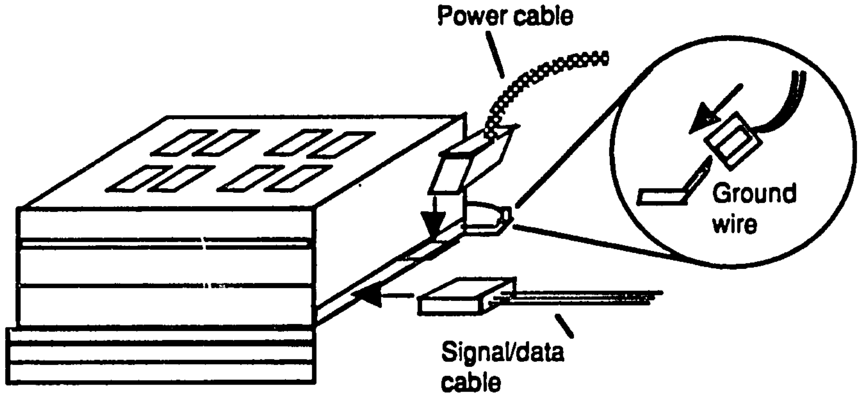
Install Disk Drives

Items already installed: motherboard; power supply

Install Floppy Drive

1. Place the floppy drive into the unit.
2. Slide the drive about two inches toward the rear of the unit, and then connect the power connector and signal connector at the top-rear of the drive (Figure 5-15).

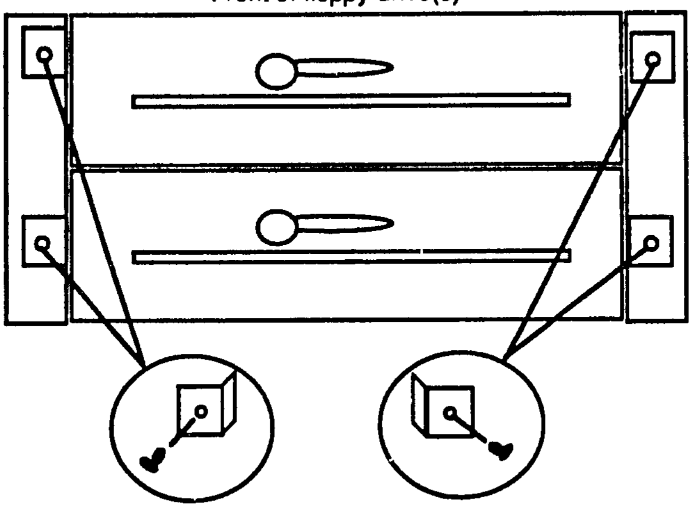
Fig. 5-15. Connect Floppy Drive Cables



3. Install floppy drive retainer hardware (Figure 5-16).

Fig. 5-16. Install Floppy Drive Retainer Hardware

Front of floppy drive(s)

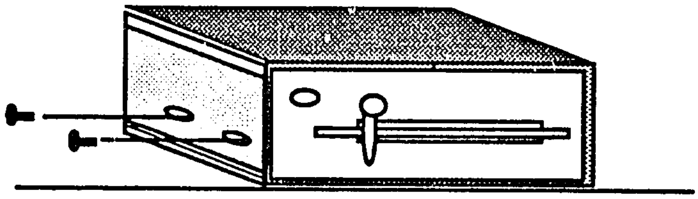


Install floppy drive retaining tabs

4. Install floppy drive mounting screws (Figure 5-17).
5. Remove scratch disk from the drive.

Fig. 5-17. Install Floppy Drive Mounting Screws

Install floppy drive mounting screws
(Note: screws sometimes are on bottom of drive)



Install Hard Drive

1. Place the hard drive into the unit.
2. Slide the drive forward about two inches toward the back of the unit.
3. At the lower rear of the drive (Figure 5-18), connect:
   - Data Connector
   - Control Connector
   - Power Connector
4. Install hard drive retainer hardware (Figure 5-19).

Fig. 5-18. Connect Hard Drive Cables

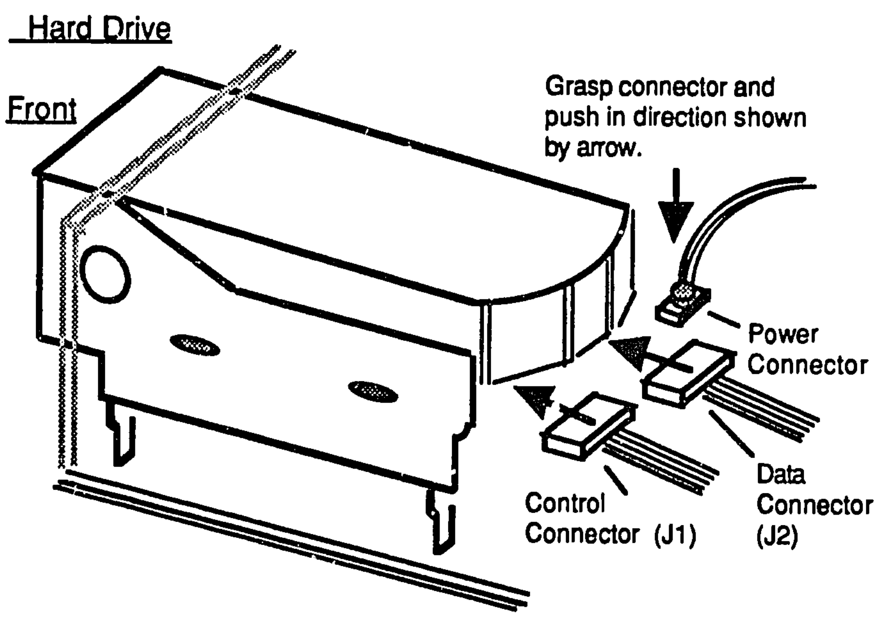
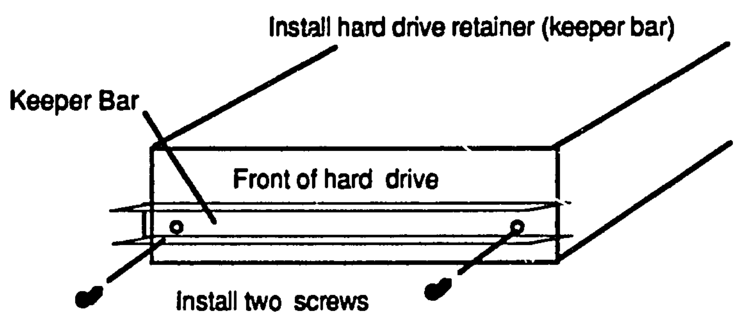
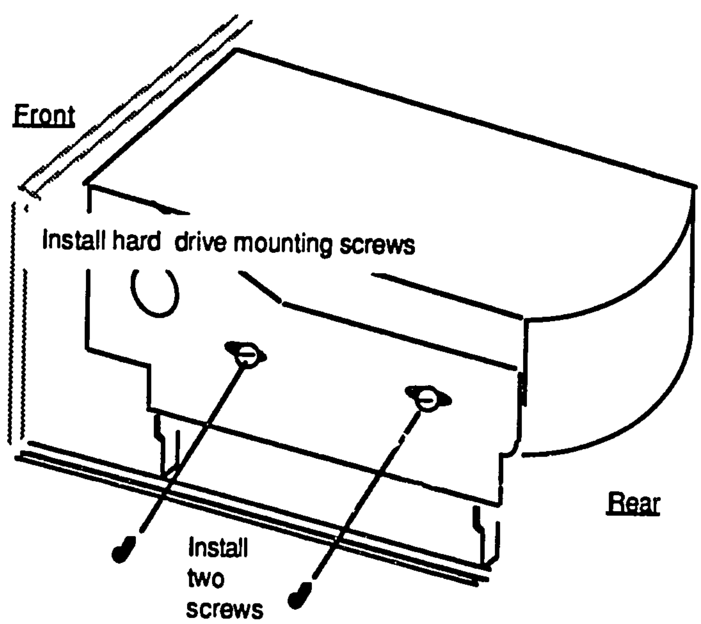


Fig. 5-19. Install Hard Drive Retainer Hardware



5. Install the hard drive mounting screws (Figure 5-20).

Fig. 5-20. Install Hard Drive Mounting Screws



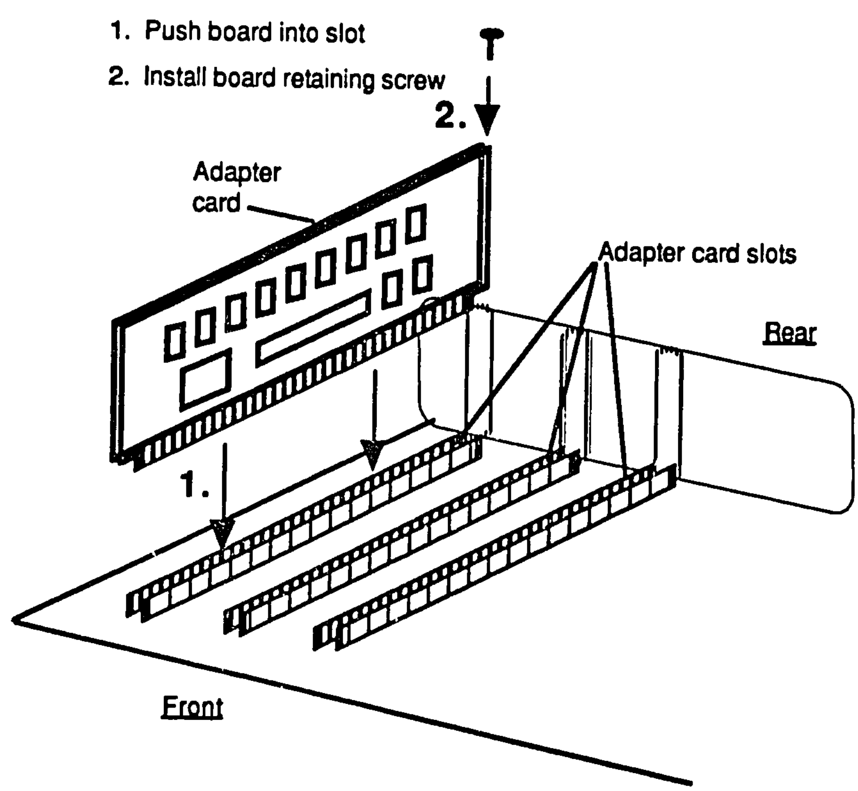
Install Adapter Boards/Cards

Items already installed: motherboard; power supply; disk drives

Refer to the sketch of the adapters that you made during disassembly. Odd-colored stripe on ribbon connector indicates pin number one (Pin #1). On the board, switches are labeled "SW" and jumpers are labeled "J."

1. To install an adapter, grasp the adapter at both ends and align edge connecter with adaptor slot; press the adapter downward, evenly, into the slot; do not twist or bend the adapter.
2. Install the retainer screw that holds the adapter in place. There is one retainer screw for each adapter (Figure 5-21).
3. Connect the adapter cables in accordance with the drawings you made during disassembly.

Fig. 5-21. Install Adapter Card Retainer Screw



Install Cover

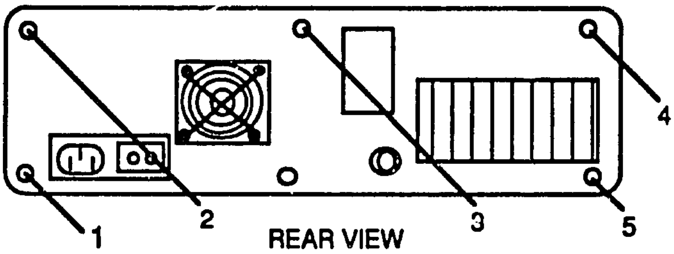
1. Place the system unit on table or workbench with adequate room to work; position so you are looking at the front panel.

2. Align the rear of the cover with the front of the unit and slide the cover onto the unit; take care not to snag any interior cables; also see that the lower edges of the cover are properly aligned when sliding the cover on.

3. At the rear panel, install five (5) retainer screws (Figure 5-22).

Fig. 5-22. Install Cover Retainer Screws

Install five (5) cover retainer screws



REAR VIEW

PS/2 SYSTEMS ASSEMBLY

Assembly instructions are presented in separate sections for:

- PS/2 Models 30 and 3-286
- PS/2 Models 50, 50Z and 70
- PS/2 Models 60 and 80

PS/2 MODELS 30 AND 30-286 ASSEMBLY

Components will be installed in the following order:

1. Motherboard (System Board)
2. Single In-Line Memory Modules (SIMMS)
3. Power Supply
4. Bus Adapter
5. Adapters
6. Hard Drive (Fixed Disk Drive)
7. 3 1/2 Inch Floppy Drive
8. Rear Panel Cover
9. Top/Side Cover

Install Motherboard (System Board)

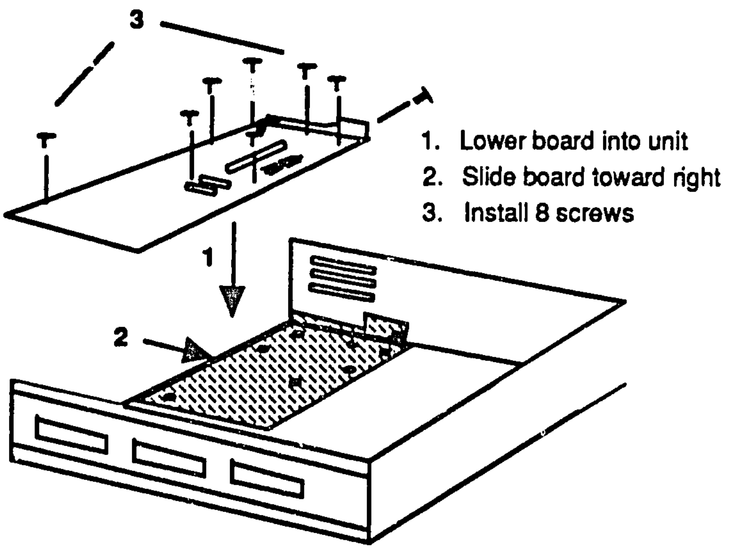
Eight screws are used to hold the motherboard in place.

1. Lower the board into the unit (Figure 5-23).

2. Slide the board sideways, toward the power supply location.

3. Install all eight screws.

Fig. 5-23. Install Motherboard



Install Single In-Line Memory Modules (SIMMS)

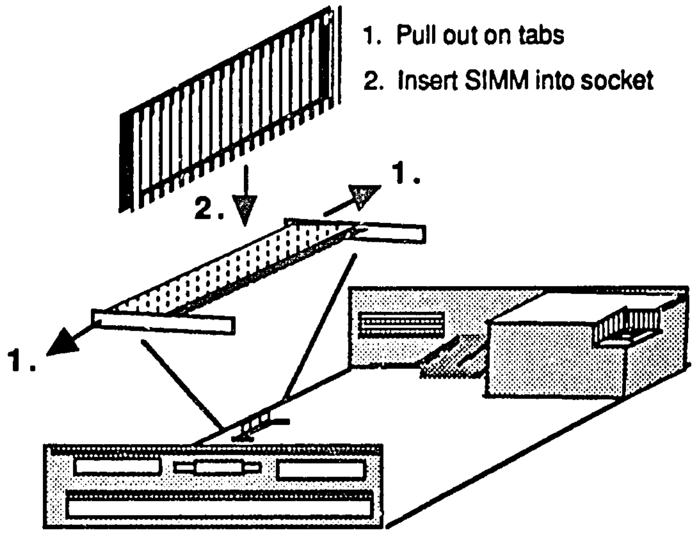
SIMMS are held in place by two tabs, one at each end of the component.

1. Gently pull outward on each tab; at the same time, carefully align the SIMM with the socket (Figure 5-24).

2. Press the SIMM into the socket.

B4

Fig. 5-24. Install SIMM



Install Power Supply

The supply is held in place by tabs underneath it that engage the bottom panel of the chassis, and by three screws through the back panel. Also, the power supply switch is attached to a link that must be engaged when installing the supply. The power supply cable is attached to the motherboard.

1. Lower the supply into the unit (Figure 5-25).

2. Slide supply toward the rear of the unit to engage the tabs beneath the supply (Figure 5-26).

3. At the back panel, install the three power supply retaining screws (Figure 5-27).

Fig. 5-25. Lower Power Supply Into Unit

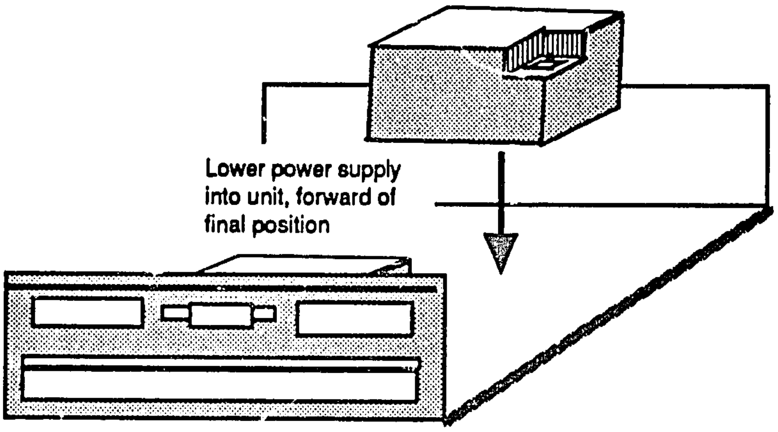
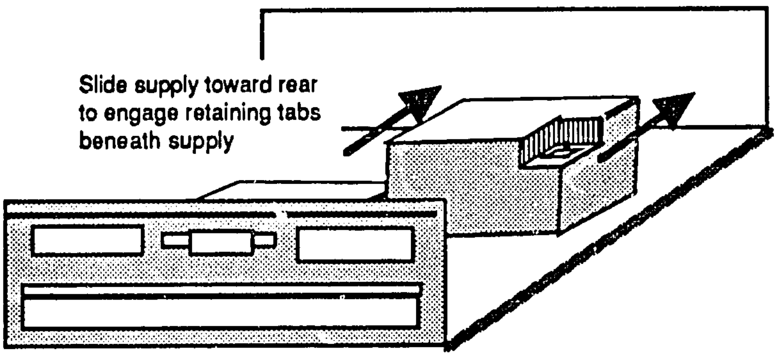
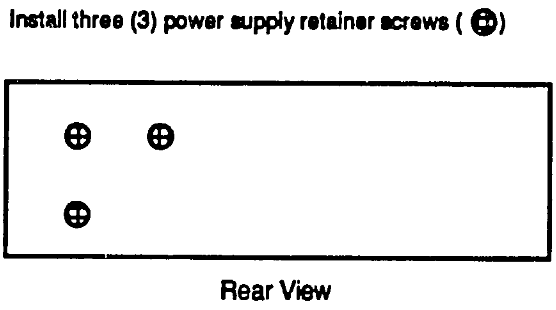


Fig. 5-26. Engage Power Supply Retaining Tabs



136

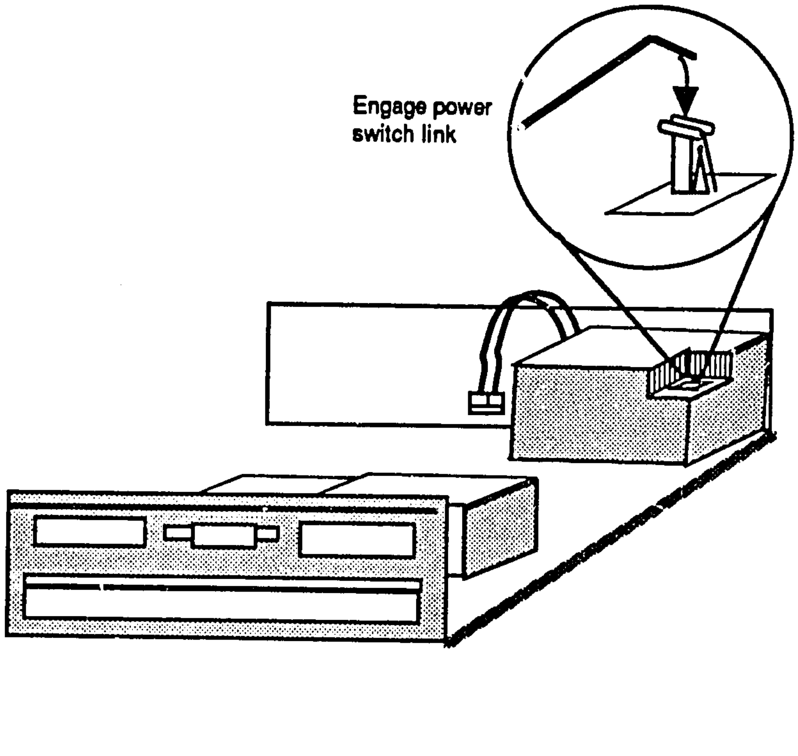
Fig. 5-27. Install Power Supply Retaining Screws at Back Panel



4. Engage the power switch link: align the link with the groove in the switch and press down (Figure 5-28).

5. Connect power supply power cable to motherboard – grasp the connector and push it straight down.

Fig. 5-28. Engage Power Switch Link



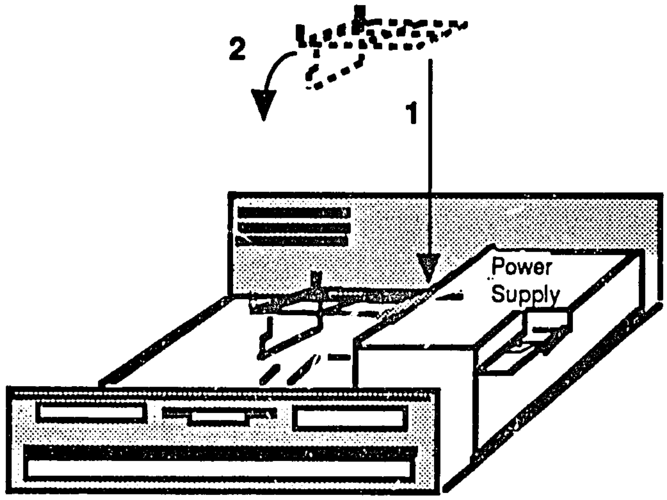
Install Bus Adapter

The bus adapter is held in place by two tabs that insert into the case of the power supply (Figure 5-29).

1. Align adapter tabs with slots in power supply case.

2. Rotate adapter downward into seated position.

Fig. 5-29. Install Bus Adapter



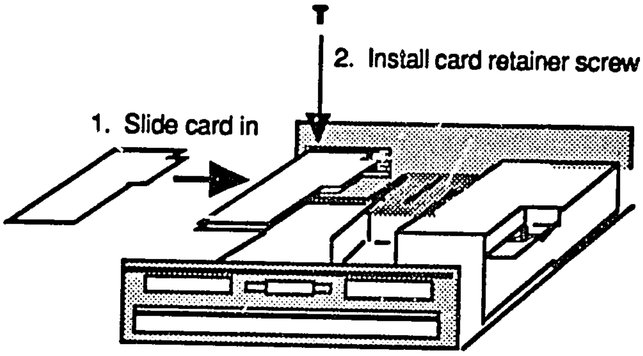
Install Adapters

The adapters slide into the unit from the side. The cards actually plug into a "bus adapter" (installed in the preceding step). Each adapter card is held in place by a retainer screw at the left rear of the card (as you face the unit from the front). The screw attaches to the card retainer bracket at the rear of the unit (Figure 5-30).

1. Grasp the card at both ends and slide it sideways, into its slot.

2. Install retainer screw at left rear of card.

Fig. 5-30. Install Adapter

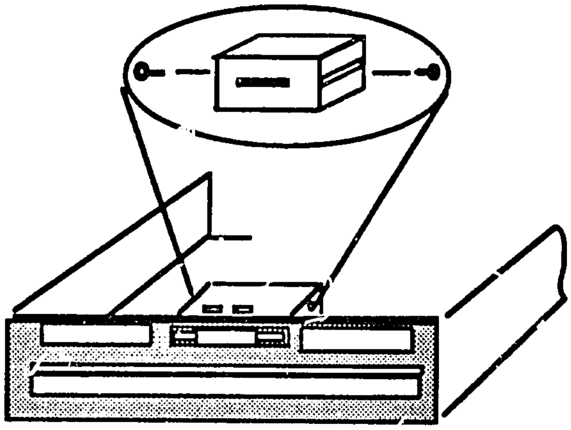


Install Hard Drive (Fixed Disk Drive)

1. Slide the drive into the front of PC; the locking tab should engage when the drive is properly positioned.

2. At each side of the drive, install the plastic nail - two nails total (Figure 5-31).

Fig. 5-31. Install Drive Mounting Nails (Plastic)

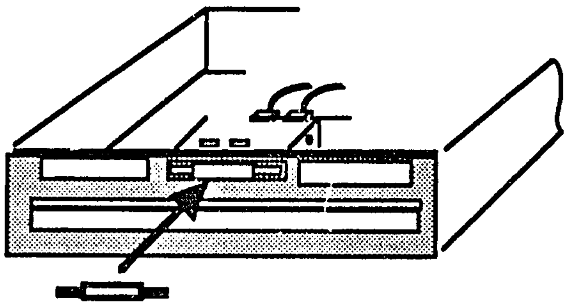


3. At the back of the drive, attach the cable connector.

4. At the front of the drive, install the plastic bezel (Figure 5-32).

5. Align the bezel and press it into position; retaining tabs will engage to hold bezel in place.

Fig. 5-32. Install Drive Bezel



Install 3 1/2 Inch Floppy Drive

1. Slide drive into front of unit; retaining tab will engage.

2. At each side of the drive, install the plastic nail (two nails total).

3. At the back of the drive, attach cable connector.

4. At the front of the drive, install the plastic bezel: align bezel and press it into position; retaining tabs will engage to hold bezel in place.

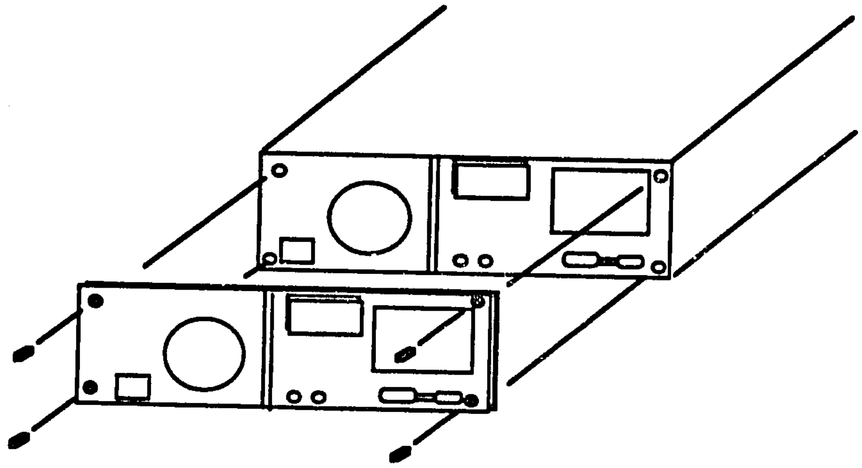
5. Remove the scratch diskette from the drive.

Install Rear Panel Cover

1. Position cover against rear panel; align to retainer screw holes (Figure 5-33).

2. Tighten four panel retaining screws – one at each corner of the back cover.

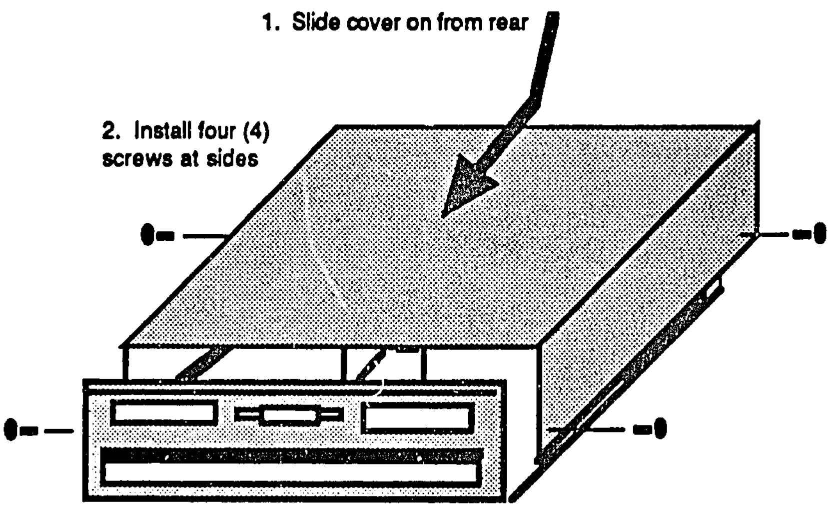
Fig. 5-33. Install Rear Panel Cover



Install Top/Side Cover

1. Align front edge of cover with rear panel of PC (Figure 5-34).
2. Slide cover toward front of PC.
3. Tighten four screws – two on the bottom corners of each side of the unit.
4. If keylock for cover is installed, lock it.

Fig. 5-34. Install Top/Side Cover



NOTES

PS/2 MODELS 50, 50Z and 70 ASSEMBLY

Components will be installed in the following order:

1. Single In-line Memory Modules (SIMMS)
2. Motherboard (System Board)
3. Power Supply
4. Drive Support Structure
5. Hard (Fixed) Disk Drive and Adapter Card
6. 3 1/2 Inch Floppy Drive
7. Adapters
8. Fan Assembly
9. Battery and Speaker Assembly
10. Cover

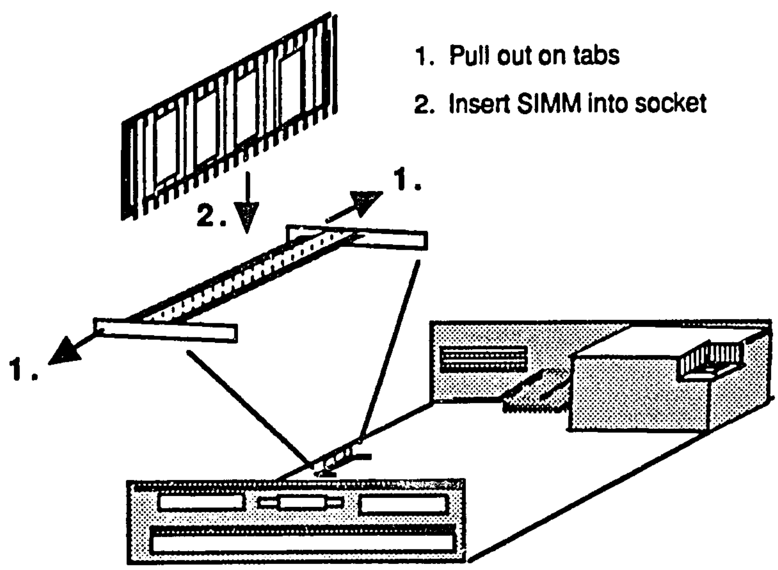
Install Single In-line Memory Modules (SIMMS)

SIMMS are held in place by two tabs, one at each end of the component.

1. Gently pull outward on each tab at the same time (Figure 3-35).

2. Insert SIMM into socket; press straight down on SIMM.

Fig. 5-35. Install SIMM

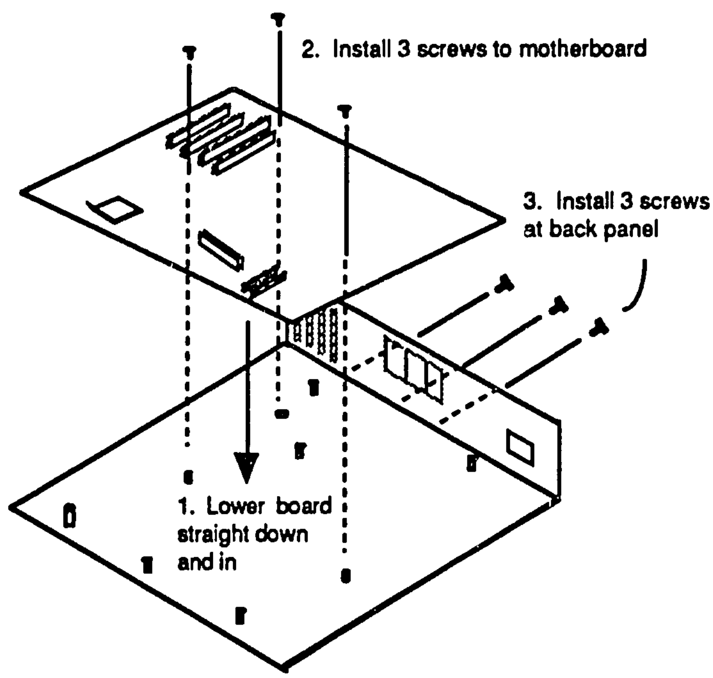


Install Motherboard (System Board)

The motherboard is held in place by six screws. Three screws are on the board, pointing downward into the bottom panel of the chassis. The other three screws are at the back panel of the unit (Figure 5-36).

1. Lower the board straight into the unit.
2. Install the three screws that hold the motherboard to the bottom panel of the chassis.
3. Install the three screws at the back panel of the PC.

Fig. 5-36. Install System Board (Motherboard)

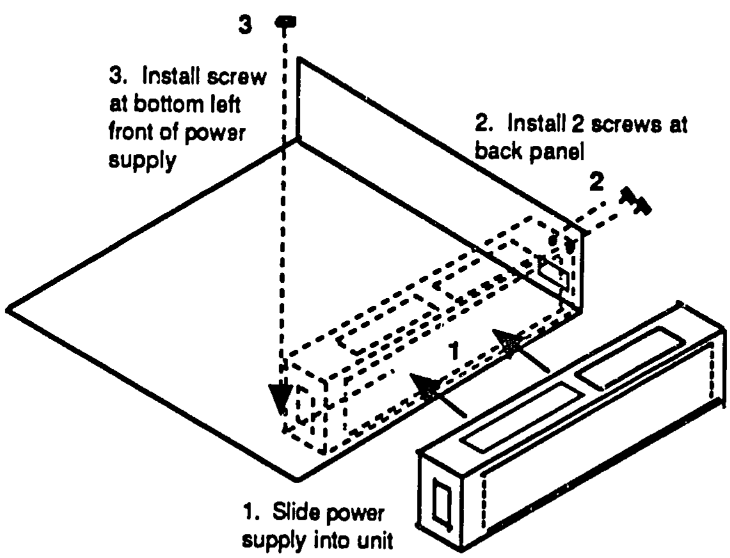


Install Power Supply

The power supply is held in place by three screws (Figure 5-37). One screw is at the front of the supply; the remaining two are at the back panel of the PC.

1. Slide the power supply sideways, out of the PC.
2. Install two screws at back of power supply (at rear panel of PC).
3. Install screw at front of power supply.

Fig. 5-37. Install Power Supply



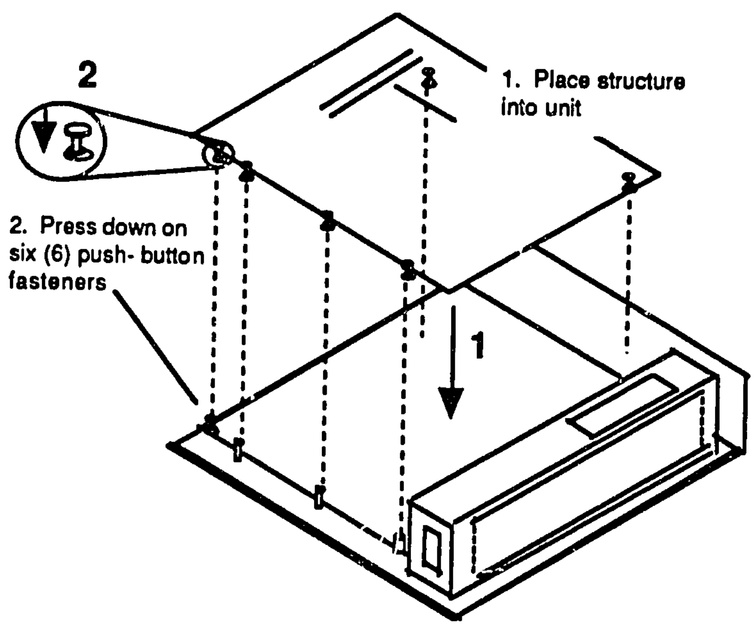
Install Drive Support Structure

The structure is held in place by six, push-button fasteners (Figure 5-38). Four of the fasteners are located at the base of the structure, toward the front of the PC. The remaining two are located toward the rear.

1. Lift the structure straight up and out.

2. Push straight down on each of the six push-button fasteners.

Fig. 5-38. Install Drive Support Structure



Install Hard (Fixed) Disk Drive and Drive Adapter Card

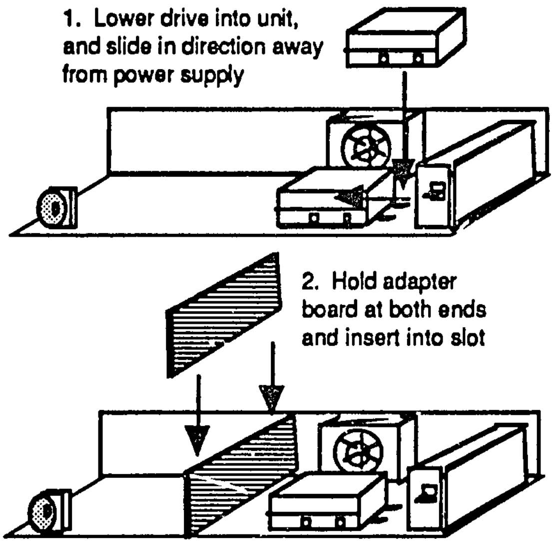
The drive is held in place by two tabs located at the base of the drive, next to the power supply (Figure 3-39).

1. Grasp the adapter at both ends and insert it into the unit.

2. Lower the drive straight down into the unit.

3. Check that retaining tabs at power supply are engaged.

Fig. 5-39. Install Hard Drive Adapter and Hard Drive



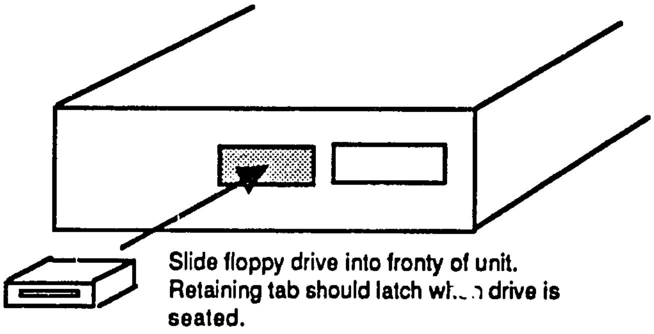
Install 3 1/2 Inch Floppy Drive

The drive is held in place by just a single tab, located at the bottom front of the drive (Figure 5-40).

1. Slide drive into its chamber at the front of the PC.

2. Check that the retaining tab is engaged.

Fig. 5-40. Install 3 1/2 Inch Floppy Drive



Install Adapters

Refer to the diagram or labels that you made during disassembly; replace adapters in the slots from which they were removed.

1. Grasp the adapter at both ends and insert it into the slot (Figure 5-41).

2. At the lower-rear corner of PC, tighten the thumbscrews (one for each adapter card) that holds the adapter card in place.

Install Fan Assembly

The assembly is held in place by two push-button fasteners, one at the top of each side of the assembly (Figure 5-42).

1. Lower the assembly straight down into the unit.

2. Press down on the two fasteners to snap them into place.

Fig. 5-41. Install Adapters

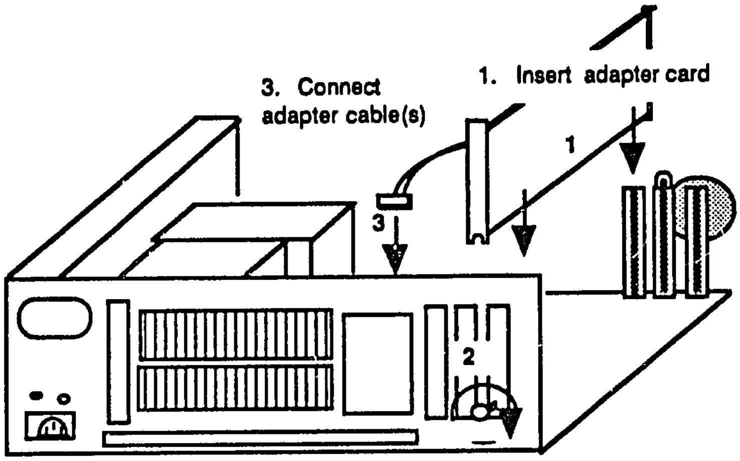
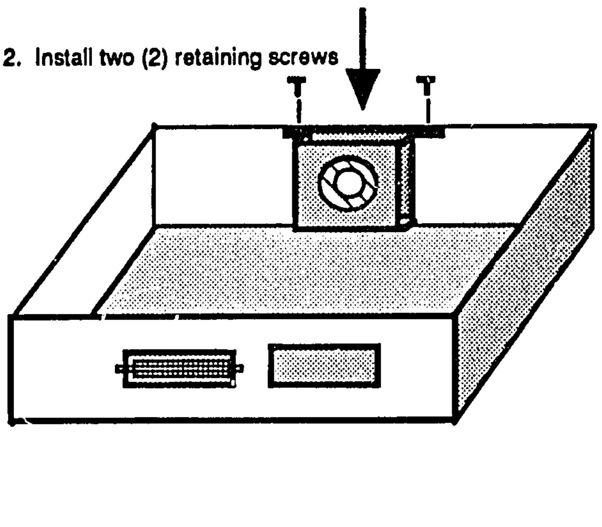


Fig. 5-42. Install Fan Assembly



150

Install Battery and Speaker Assembly

The assembly is located at the left front corner of the PC (Figure 5-43). The battery and speaker are assembled into a single unit, but the battery may be installed by itself. The battery should be installed after the assembly is installed. This will avoid accidentally shorting out (discharging) the battery as you work.

Speaker Assembly

1. Grasp assembly and insert it into the unit.

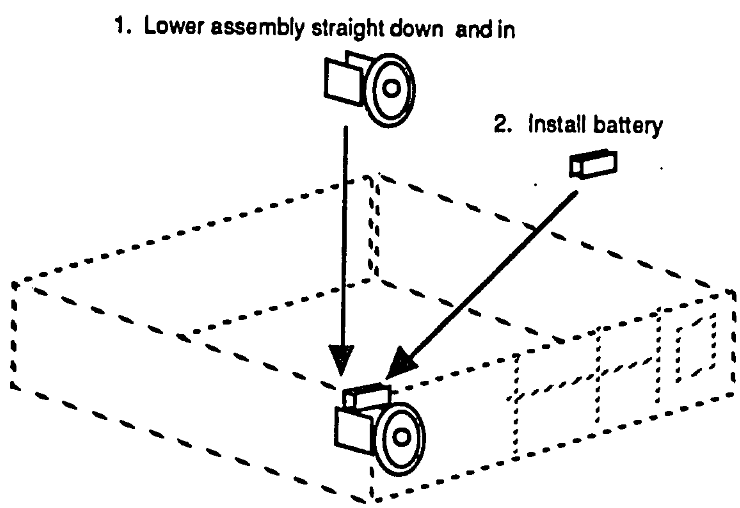
2. Take care not to damage the speaker cone; at the bottom of the assembly, the tab should engage to hold the assembly in place.

Battery

1. Grasp the battery and push it straight down and into position.

2. Retainer tabs should engage to hold battery in place.

Fig. 5-43. Install Speaker and Battery Assembly

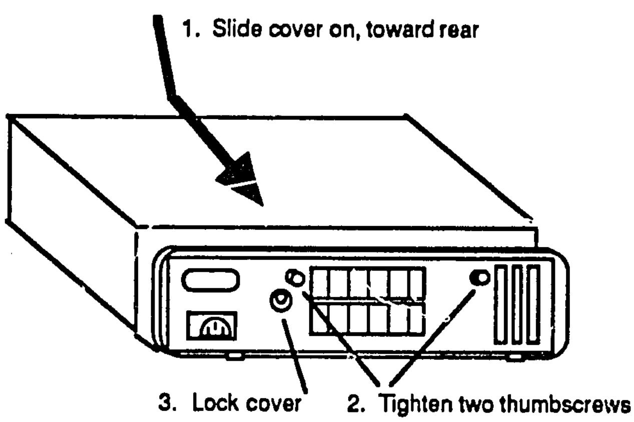


Install Cover

The cover is held in place by two thumbscrews near the top of the rear panel (Figure 5-44). After installation, the cover is secured by a keylock.

1. Align the rear edge of the cover to the front edge of the chassis.
2. Slide the cover toward the rear of the PC.
3. At the rear panel, near the top, tighten two thumbscrews.
4. Lock the cover keylock

Fig. 5-44. Install Cover



NOTES

PS/2 MODELS 60 and 80 ASSEMBLY

Components will be installed in the following order:

1. Motherboard (System Board)
2. Hard (Fixed) Drive Support Structure
3. Hard (Fixed) Drive C
4. Hard (Fixed) Drive D
5. Floppy Drive Cable Retainer
6. Floppy Drives
7. Power Supply
8. Front Bezel
9. Battery and Speaker Assembly
10. Adapters
11. Cover

Install Motherboard (System Board)

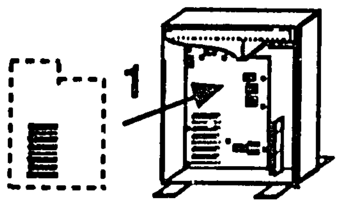
The motherboard is held in place by eight retaining screws. The board is connected to the unit by cables.

1. Place the board into the unit so as to align it with the retaining screw holes (Figure 5-45).

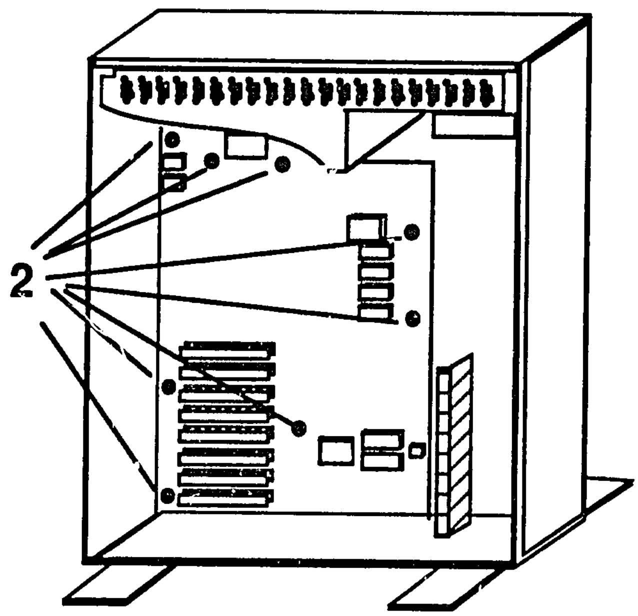
2. Install the eight retaining screws.

Fig. 5-45. Install Motherboard

1. Position board inside unit



2. Install 8 retaining screws (●)



155

Install Hard (Fixed) Drive Support Structure

A metal structure ("cradle") is used to support the hard drives. The structure is held in place by four screws (Figure 5-46).

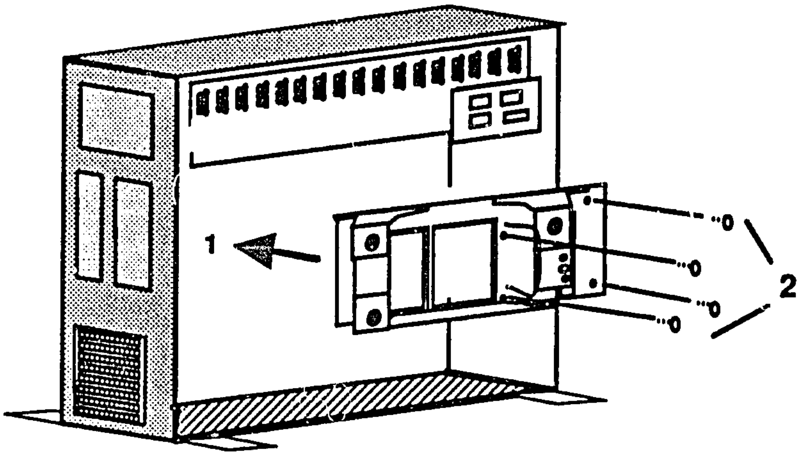
1. Place the structure into the unit, positioned slightly toward the front of the enclosure.

2. Slide the structure sideways (toward the rear of the unit).

3. Install the four retaining screws.

Fig. 5-46. Install Drive Support Structure

1. Slide structure into unit
2. Install 4 screws



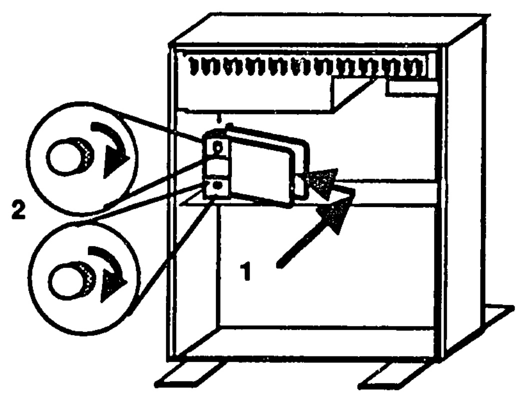
Install Hard (Fixed) Drives C and D

Because of the way the drives are physically positioned in the system, drive C must be installed before installing drive D (if present). The drives are connected to the system by cables and a ground wire. Each drive is held in place by two thumbscrews.

1. Install Hard (Fixed) Drive C (Figure 5-47).
2. Slide drive into drive support structure.
3. Tighten the two thumbscrews (turn clockwise).
4. Connect all cables and ground wire.

Fig. 5-47. Install Fixed Drive C

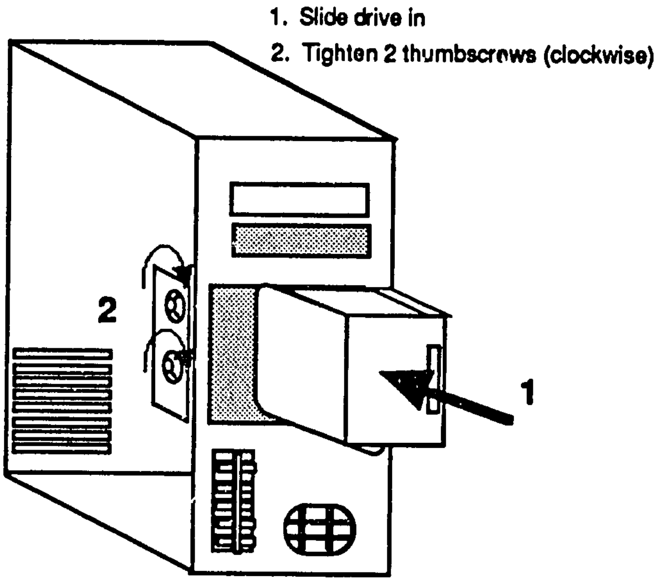
1. Position drive inside unit, slide toward back of cabinet
2. Tighten 2 thumbscrews (clockwise)



Install Hard (Fixed) Drive D (If Present)

1. Slide the drive into the drive support structure; position it slightly toward the front of the system (Figure 5-48).
2. Slide the drive slightly toward the rear of system .
3. Tighten the two thumbscrews (turn clockwise).
4. Connect all cables and the ground wire.

Fig. 5-48. Install Fixed Drive D (If Present)



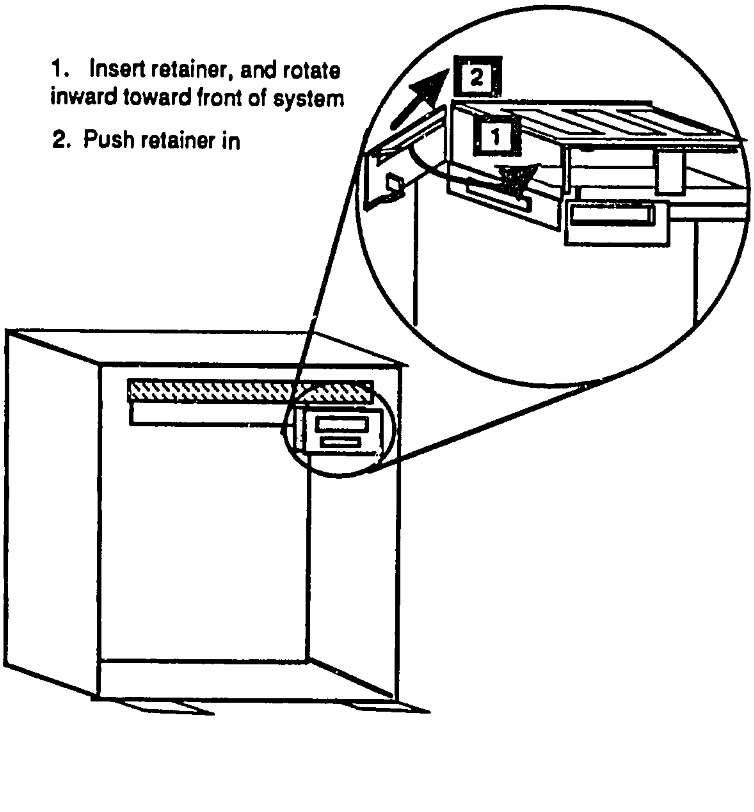
158

Install Floppy Drive Cable Retainer

When the floppy drives are plugged in, their cables are held in place by a retainer (Figure 5-49). The cable retainer itself is held in place by two tabs.

1. Position the retainer so it is angled slightly forward.
2. Rotate the retainer toward the front of the system.
3. Tabs should engage to hold retainer in place.

Fig. 5-49. Install Floppy Drive Cable Retainer



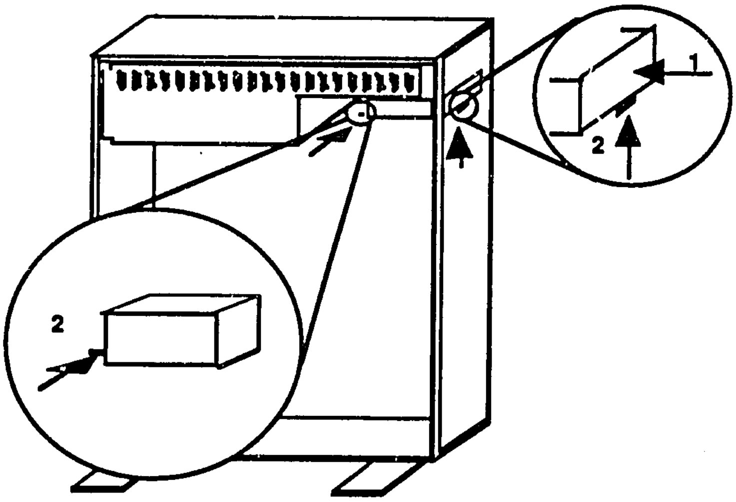
Install Floppy Drives

The drives are held in place by two catches (Figure 5-50). One (usually colored black) is at the back of the drive. The other is at the lower front of the drive.

1. Slide the drive into the unit; take care to align the back of the drive with the retaining catch that engages at the back of the drive.
2. The tab located at the rear of the drive should "click" when it engages the drive.
3. The tab at the front of the drive will snap into place when drive is fully seated.

Fig. 5-50. Install Floppy Drive

1. Slide drive in
2. Press up on tab at front, and press inward on tab at back at the same time
3. Seat drive with tabs engaged



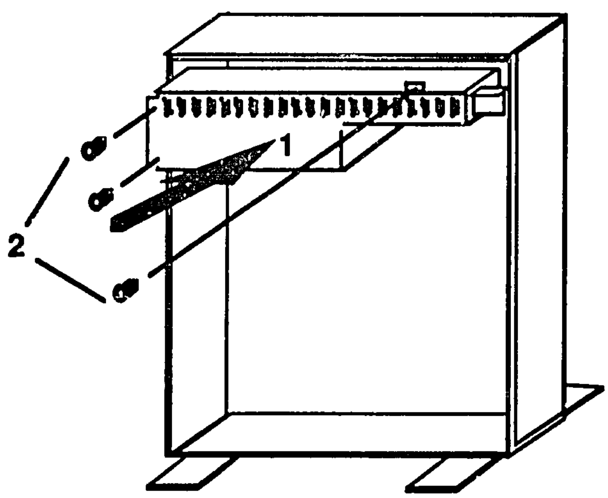
Install Power Supply

The supply is held in place by three screws (Figure 5-51). One screw is located toward the upper front of the supply, near the power switch. The remaining two screws are toward the back of the PC. As you face the side of the PC, these two screws are at your left. The power supply connects to the unit with cables.

1. Carefully support the supply from the bottom (to avoid dropping it), and slide it into the unit.
2. Connect all cables from the power supply to the system.
3. Install the three retaining screws.

Fig. 5-51. Install Power Supply

1. Slide power supply in
2. Install 3 screws



161

Install Front Bezel

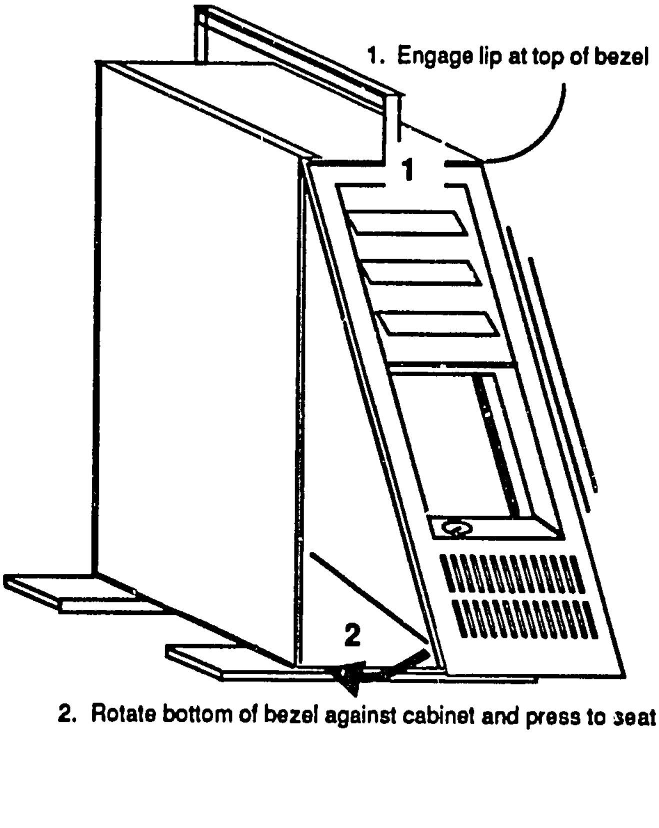
The bezel is the plastic frame (cover; panel) that protects the front of the PC. It snaps on and off (Figure 5-52).

1. Grasp the bezel and position yourself in front of the PC.

2. Tilt the top of the bezel toward the PC and align its upper edge with the upper edge of the PC front panel.

3. Hold the bezel against the PC and apply pressure down the sides, rotating the bottom of the bezel toward the PC front panel; the bezel should snap into place.

Fig. 5-52. Install Front Bezel



Install Battery and Speaker Assembly

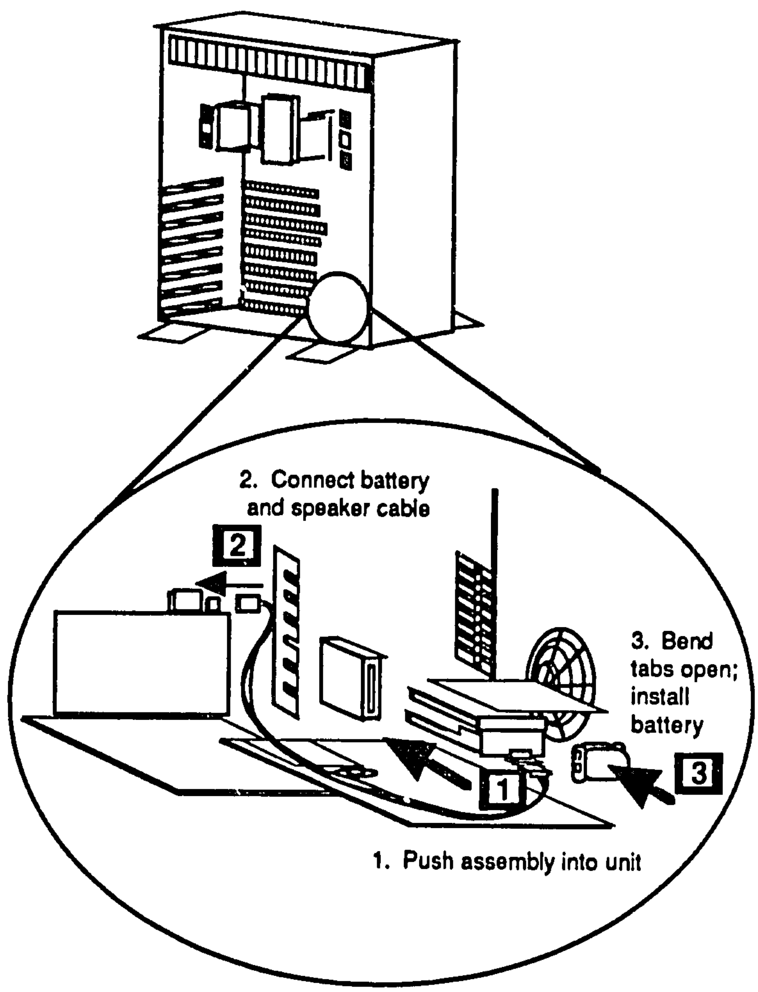
As you face the PC from its left side, the assembly will be installed in the lower right hand corner of the enclosure (Figure 5-53). The battery and speaker are assembled into a single unit, but the battery may be installed by itself. The battery should be installed after installing the assembly. This will avoid accidentally shorting out (discharging) the battery as you work.

1. Insert the assembly into position; the tab should engage at bottom of assembly.

2. Connect the battery/speaker cable.

3. Insert the battery; tabs should engage to hold battery in place.

Fig. 5-53. Install Battery and Speaker Assembly



607

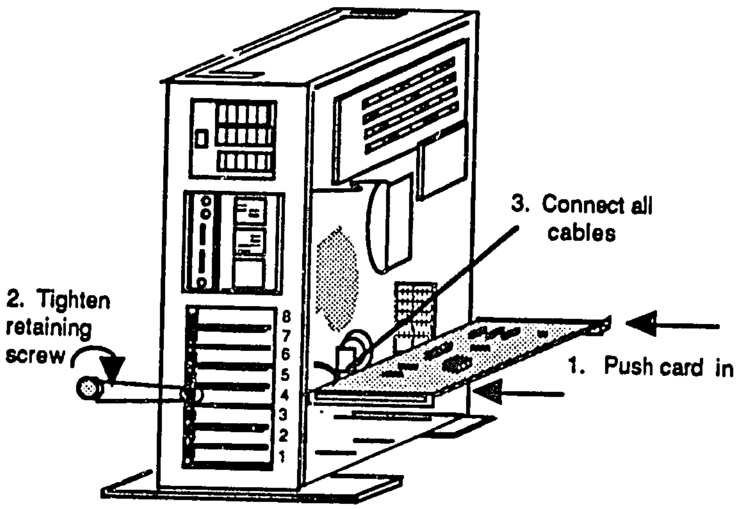
Install Adapters

Refer to the diagrams and notes you made during disassembly. Install adapters into the slots from which they were removed.

1. Grasp adapter at outer edges, and push straight into adapter slot (Figure 5-54).

2. Connect all cables to the adapters.

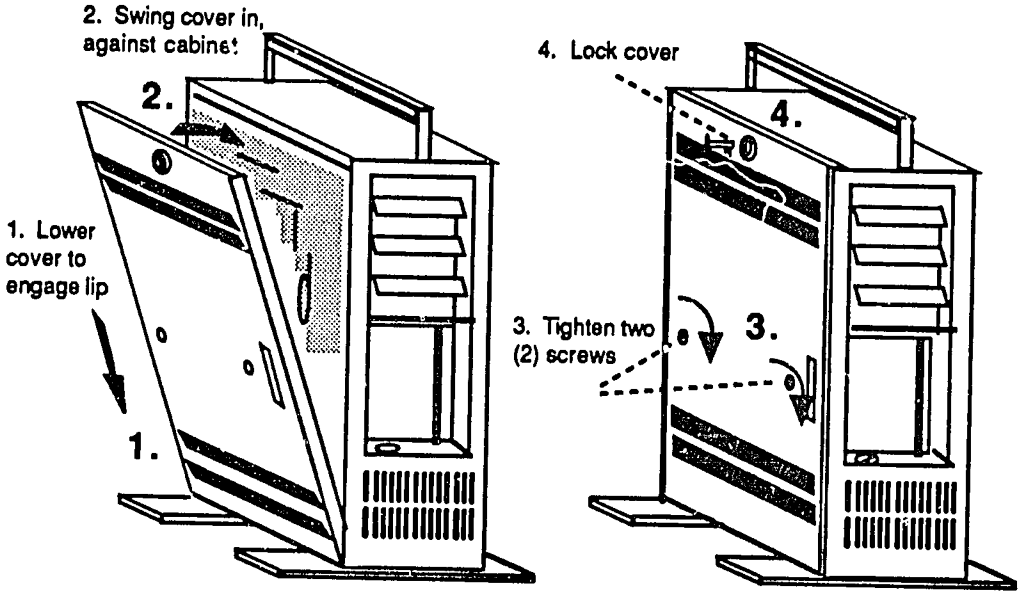
Fig. 5-54. Install Adapters

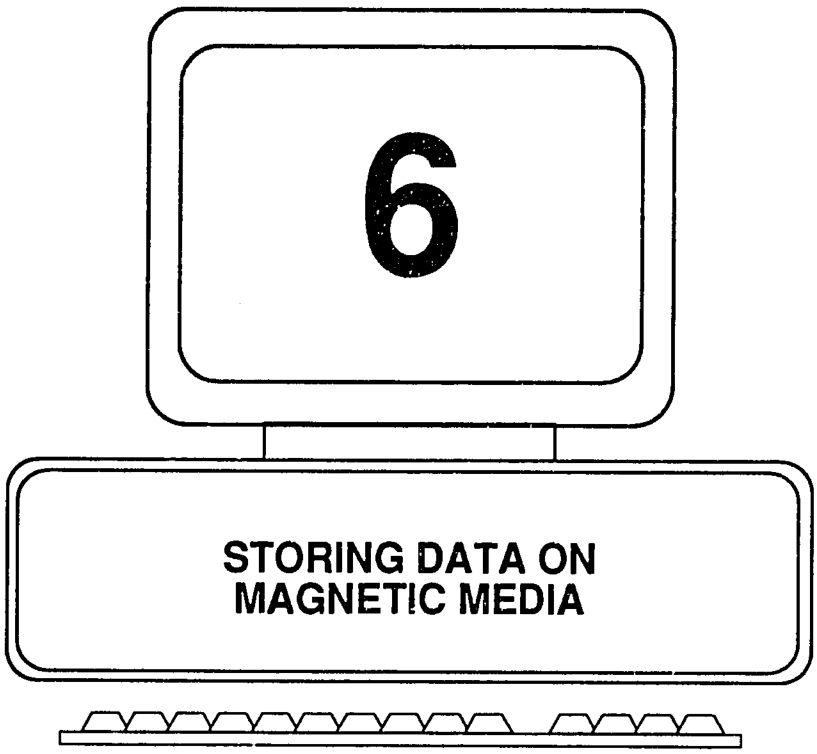


Install Cover

1. Tilt the bottom of the cover toward the PC, and seat the lower edge of the cover (Figure 5-55) into the base of the cabinet.
2. Rotate the cover toward the cabinet, into position.
3. At the side of the enclosure, tighten two thumbscrews (clockwise).
4. Lock the cover keylock.

Fig. 5-55. Install Cover





6

STORING DATA ON MAGNETIC MEDIA

CHAPTER 6: STORING DATA ON MAGNETIC MEDIA

In this chapter, we will work our way through the world of floppy disks and hard disks, starting with the fundamentals of magnetism and magnets. We will see how magnetic charges are used to store information on disks, how a disk is prepared (formatted) to receive information, and how the contents of a disk are kept track of by the system. We will look briefly at how the platters (disks) of a hard drive interact with the drive read/write heads in order to store and retrieve data. We also will discuss tracks, sectors, cylinders, clusters (file allocation units) and the File Allocation Table (FAT).

Magnets

There are two basic types of magnets – permanent magnets and electromagnets. Both play important roles in storing data on and retrieving data from magnetic media (disks, tape), and in the operation of impact (e.g., dot matrix) printers and laser (photocopy) type printers. A basic understanding of magnetism should be helpful in learning how these devices operate and in troubleshooting them when problems arise.

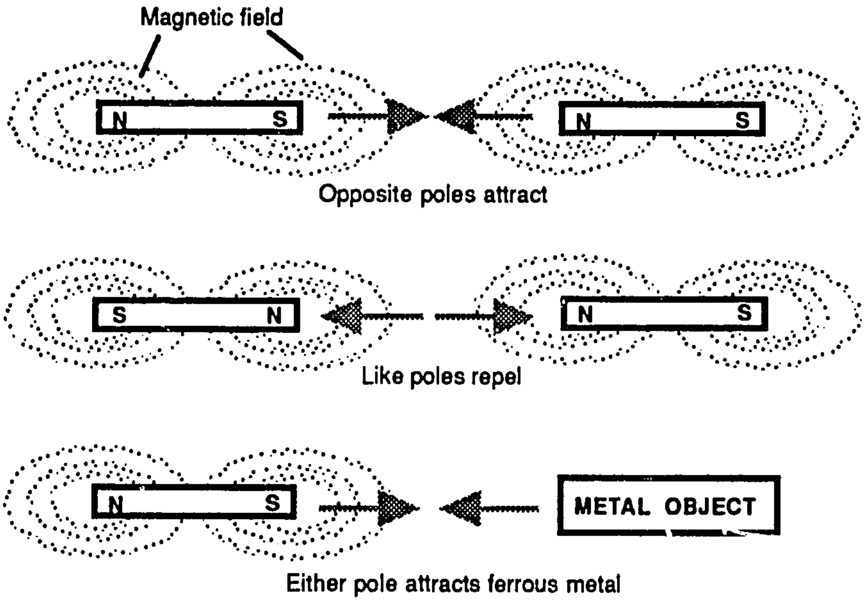
Permanent Magnets

A magnet is piece of metal that can attract another piece of metal. Note that in all cases, we mean those metals that have the properties of magnetism. Iron is one, for example, and nickel is another; but stainless steel and aluminum are not.

A permanent magnet is one that retains its magnetic properties without the aid of any external electrical power. As shown in Figure 6-1, a magnet has two "poles," one positive, the other negative (usually labeled N for north, and S for south). Either pole will attract a non-magnetized piece of metal (a non-magnet). Given two magnets, however, the opposite poles will attract each other (positive and negative will attract each other, and the similar poles will repel each other (positive will repel positive and negative will repel negative). Permanent magnets are found commonly in households and offices, where they are used, for example, to "stick" things to a refrigerator and to hold paper clips.

A magnet is created by exposing a piece of magnetizable material (e.g., iron or iron oxide) to a strong magnetic force. The force may be provided by another magnet or from the electromagnetic field generated when electrical current flows through a conductor (e.g., a coil of wire). This principle is used to create magnetic spots on the surface of a disk, and thus to "write" data to the disk. The same principle, applied in reverse, is used to "read" data from the disk.

Fig. 6-1. Positive and Negative Poles of a Magnet



Electromagnets

An electromagnet is basically a piece of metal, surrounded by a coil of wire. While electrical current flows through the wire coil, it causes the piece of metal to becomes magnetic. When the flow of current stops, the piece of metal is no longer magnetic (or it is at least considerably less magnetic). It is also true that if you place a magnet and an electrical conductor (e.g., wire) in proximity and move one in relation to the other, the magnet will induce an electrical current to flow in the wire. The principle is known as Faraday's Law, illustrated in Figure 6-2.

By applying electrical current intermittently to the coil, one can "turn the magnet on and off" at will. Since a magnet can exert a force on a metal object, it is possible to use magnetism to physically move an object. Thus you can open and close a switch by "remote control," as illustrated in Figure 6-3. Or, you can cause the pins on a dot matrix printer to move, striking the ribbon and thus printing information on paper, as depicted in Figure 6-4. The same principle is used to move the stepping motor of a disk drive, to position the read-write heads (which are themselves electromagnets, as explained shortly).

168

Fig. 6-2. Faraday's Law

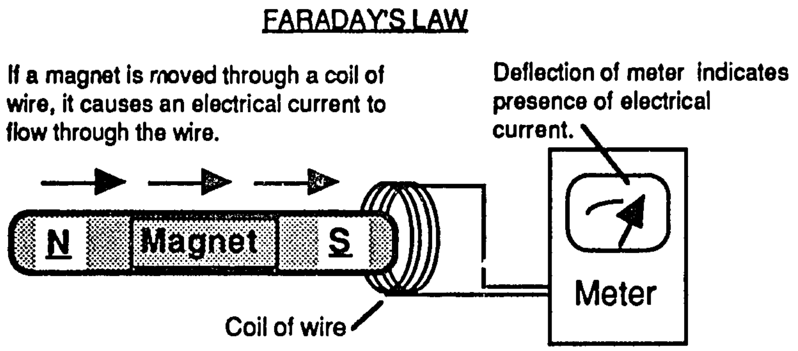


Fig. 6-3. A Switch Controlled by an Electromagnet

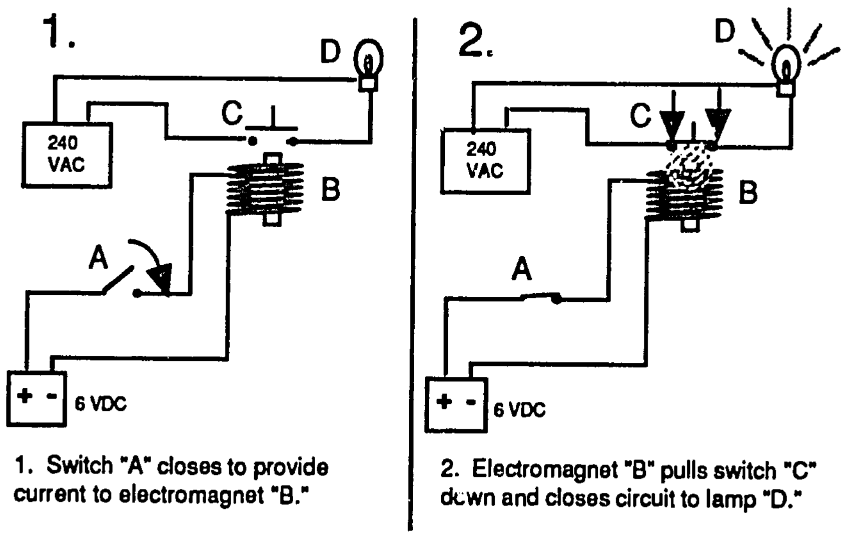
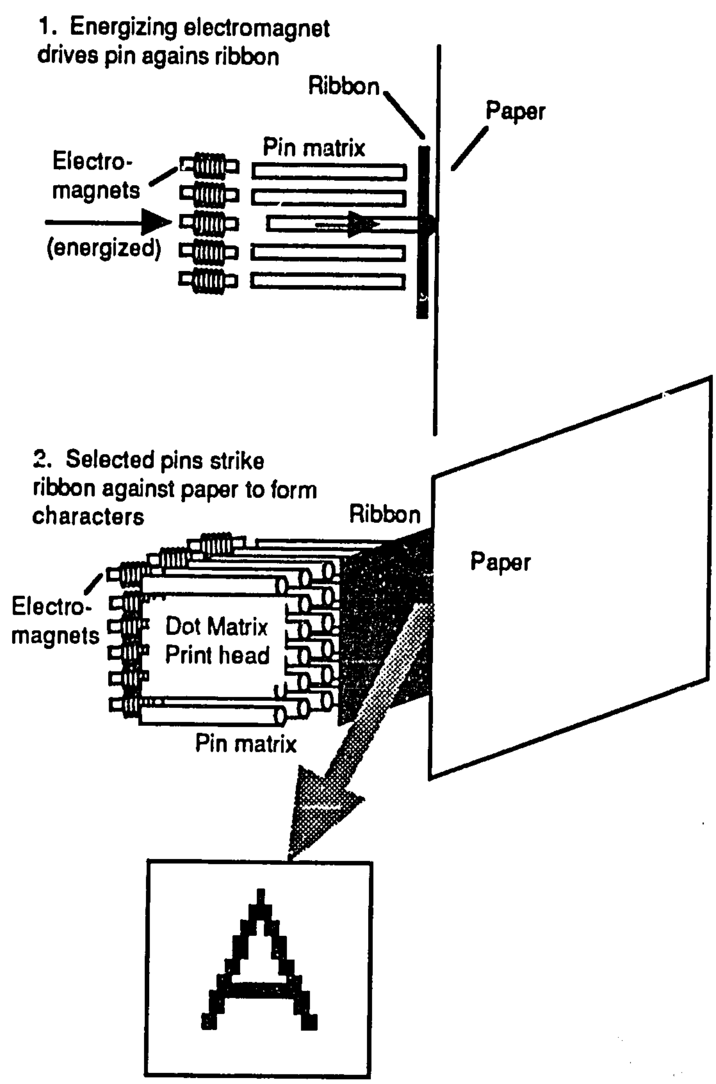


Fig. 6-4. Electromagnetism Operates a Dot Matrix Printer Head



170

Magnetic Writing on a Disk

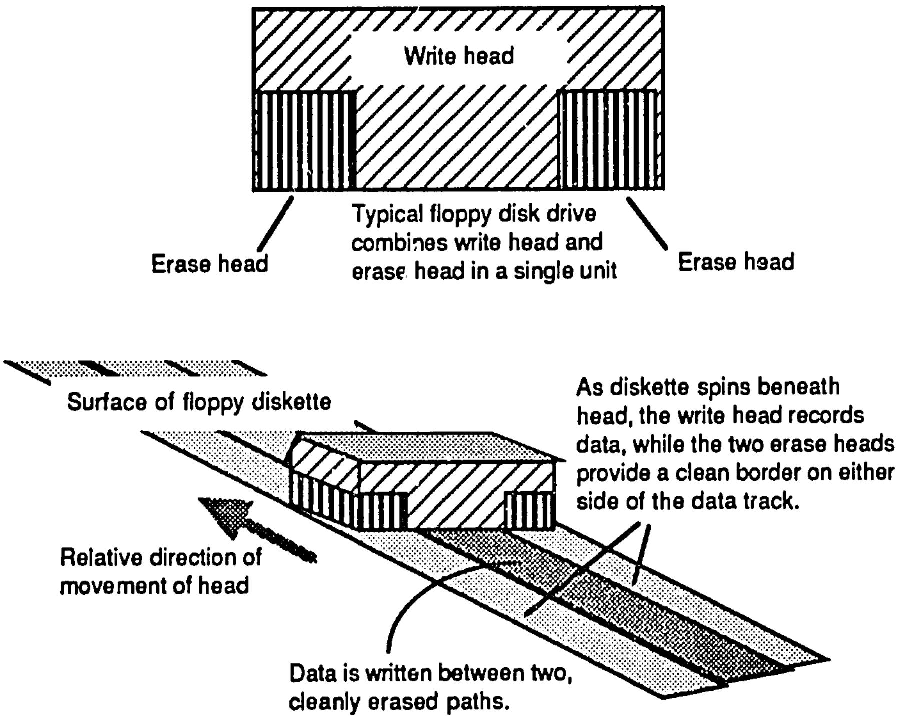
If a magnetizable surface, such as a storage disk, is revolving beneath an electromagnet, it is possible to place magnetic charges at various points on the disk, i.e., to "write" data. The device for doing this is called a "write head" and it is essentially an electromagnet. Whereas the property of permanent magnetism is used to store data on disks, the "electromagnet" is the key to writing the data to the disk in the first place, and later reading it from the disk. Figure 6-5 illustrates how the read, write and erase heads create a strip of magnetized points within a track on a data disk. Notice how the erase heads "trim" the edges of the strip to provide a clean line of demarcation between adjacent data tracks. Of course, the strip is not just a continuous random stream of magnetized particles representing data on the disk. The data is stored according to a pattern of microscopically small, magnetized spots as explained below.

Note that although we speak of a disk as containing "documents and data," the letters, words and punctuation marks of a document are never stored as alphabetical characters and symbols. Instead, each character and symbol is assigned an "identification number" and that number written to the disk as a string of 1s and 0s. It is the job of the computer to translate the letters (keystrokes) we type at the keyboard into their respective identification numbers, store the numbers, and later retrieve and translate them back into characters that have meaning to us.

Thus, the surface of the media is, in effect, a collection of tiny permanent magnets. The device that places the magnetic charges on the disk is called a "write head." The device that later senses the presence of the magnetic charges to retrieve information from the disk is called a "read head." Still another device, an "erase head" is used to alter the magnetic charges so that they are unreadable or, in effect, "mean nothing" to the computer in the sense that the magnetic condition at that point does not represent data or other information of use. In actuality, the read head, write head and erase head all are combined into a single unit. Which one is active at any moment (to read, write, erase) is controlled by the electronic circuitry of the computer.

The surface of the magnetic media (disk or tape) is coated with iron oxide. Iron oxide is commonly known as "rust" – it is iron that has combined with oxygen, much the way rust forms on your car. Even though it is not pure iron, it still has the properties of iron insofar as magnetism is concerned. If the iron oxide is exposed to a magnet, the iron oxide itself will become a magnet So, if we expose the disk surface to a magnet, the exposed area will become magnetized, forming a kind of magnetic "trail" on the disk, as illustrated in Figure 6-5. This is how we "write" data onto a disk. The magnet that we place next to the disk is called a "write" head.

Fig. 6-5. Magnetic "Trail" Left by Write and Erase Heads



The magnetic trail as shown in Figure 6-5 bears closer inspection. It is not, as it might first appear, simply a stripe or swath left by a kind of magnetic paint brush. As the disk spins beneath the write head, the head is being turned on and off at a very high rate. It thus creates a pattern of magnetically charged spots interspersed with spots that are not so charged. You can think of the charged spots as an arrangement of tiny magnets across the disk surface. As will be explained shortly, the presence or absence of a magnetized spot can be interpreted to represent a "yes" or "no" condition or, correspondingly, as the number 0 or 1.

Representing Information Magnetically

Now let's see how these patterns of magnetic spots are used to represent data. Assume that we agree ahead of time on all of the spots where a magnetic charge could be placed. Then, we magnetize some of the spots, and leave some others non-magnetized. Next, we check each spot, one after the other in a systematic order, and write down our results. To keep things simple, we'll deal here with only 8 spots, arranged in a row. We will agree that if a spot is magnetized, we will write down a 1, and if it is not, we will write a 0.

If none of the spots were magnetic, the results would be:

0 0 0 0 0 0 0 0

If all of the spots were magnetic, the results would be:

1 1 1 1 1 1 1 1

A combination of magnetized and non-magnetized spots might produce something like the following result:

1 0 1 1 0 0 1 1

or

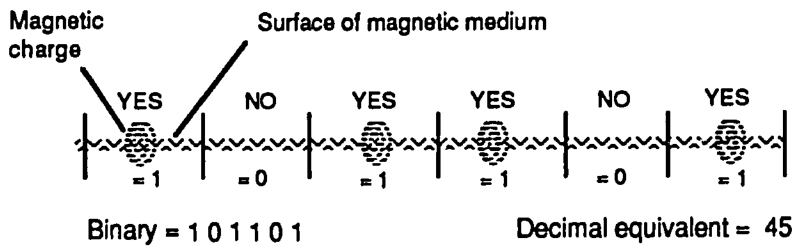
1 1 0 1 0 1 1 0

or

1 1 1 1 0 0 0 0

Figure 6-6 illustrates how the magnetic spots are assigned values of "yes" or "no," or binary values of "1" or "0." In the figure, the string of zeros and ones shown represents the binary number 101101. The decimal equivalent is 45. For a discussion of decimal, binary and hexadecimal numbers, please refer to Chapter 8.

Fig. 6-6. Write Head Produces a Pattern of Magnetized Spots on the Disk Surface



In fact, using just 8 spots there are 256 sets of results possible – all zeros, all ones, and all combinations of zeros and ones in between. With just 8 tiny spots on the disk, we are able to represent a total of 256 different conditions (i.e., 256 possible combinations of 0s and 1s), by either magnetizing or not magnetizing a given spot, as shown in Figure 6-7. The strings of 0s and 1s are in fact binary numbers, and each binary number, of course, can be translated directly into its decimal equivalent.

Since the disk in effect becomes a set of tiny permanent magnets, it is able to store the data without being supported by external electrical power. Later, when a "read head" is exposed to the magnetic fields of the disk, the read head is influenced such that it can "report" the presence of the fields on the disk and thus retrieve data from the disk – Faraday's Law in action once again.

The total of 256 works out just fine, because we need a distinct number to represent each of the digits 0-9, the 26 lower case letters of the alphabet, the 26 upper case letters of the alphabet, all of the other symbols on the keyboard, and some special characters. This turns out to be 256 items in all. Of course, with our meager 8 bits, we can only represent one number or character at a time, but it's a start.

174

Fig. 6-7. Using 8 Spots on a Disk to Represent Numbers

Possible Result (Binary)	Decimal Number
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
00000101	5
00000110	6
00000111	7
00001000	8
00001001	9
00001010	10
00001011	11
00001100	12
00001101	13
...	...
11111111	255

Bits and Bytes

Since the basic set of characters that we are interested in is 256 in total number, and since its takes 8 bits to represent the range of numbers 0-255 (so that each digit and character may be assigned its own "identification number"), a set of 8 bits is used as the unit to represent a digit, symbol, or character. Just as we ordinarily express travel distance in miles rather than feet or yards to avoid dealing with inconveniently large numbers, so do we represent information storage capacity in units of 8 bits. In computer parlance, 8 bits is called a "byte."

The byte is a unit of practical value in that it represents the capacity to store a single character of the alphabet, a digit, a symbol, and so on. Now we can see that it takes 1 byte (8 bits) to store a single letter. Assuming there are, on average, 8 letters per word and 250 words per typewritten page of text, we can calculate the number of bytes it takes to store a page of text: 1 byte per letter x 8 letters per word x 250 words per page = 2,000 bytes per page of text.

Kilobytes and Megabytes

Thousands of bytes are expressed in kilobyte units, where "kilo" means "thousand." kilo is abbreviated as "k." Millions of bytes (b) are expressed in megabyte units, where "mega" means "million." In ordinary mathematical expressions, mega is abbreviated as "m." Since the binary system is based on powers of 2, the capital letters K and M are used to indicate 1024 byte units and 1,048,576 byte units respectively. Thus, 1kb would represent 1000 bytes, whereas 1K represents 1024 bytes. Similarly, 1m represents 1 million bytes, whereas 1Mb represents 104,857,600 bytes. Thus, a disk capacity designated as 100Mb indicates not 100 million bytes, but rather 1,048,576 bytes.

In Figure 6-8, the right hand column shows the value obtained by raising 2 to the exponent (power) shown in the left hand column. Any number raised to the power zero is, by definition, 1. Therefore $2^0 = 0$. Any number raised to the power 1 is he number itself. So, $2^1 = 2$. From there on, the value increases:

$2^2 = 2 \times 2 = 4$
$2^3 = 2 \times 2 \times 2 = 8$
$2^4 = 2 \times 2 \times 2 \times 2 = 16$

and so on.

Recall from our example using 8 spots that a string of 8 binary digits (0s and 1s) allowed us to represent 256 characters or numbers. In Figure 6-8 we can now see why. Two raised to the eighth power (2^8) = 256. Remember that a string of "all zeros" counts as a number. In Figure 6-8, the highest decimal number shown is 255, but note that we began counting from zero. This gives us a total of 256 numbers. Note also in Figure 6-8 that $2^{10} = 1024$, from whence comes the definition of 1Kb, as discussed earlier.

Typical PC hard disk systems today store from 40M bytes to 120M bytes or more. Considering that each of the 120M bytes requires 8 bits (8 individual magnetic charges), we can see that these devices require very precise design and a proper operating environment. It also suggests that there are many ways for things to go wrong on a hard drive, and they often do – especially for those who do not regularly backup their data.

Now that we understand how information is stored on a disk by using magnetic charges, let's see how a disk is prepared to receive the information. The process is called "formatting."

Fig. 6-8. Powers of 2

Exponent (x)	2^x
0	1
1	2
2	[illegible]
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1,024
11	2,048
12	4,096
13	8,192
14	16,384
15	32,768
16	65,536

Formatting

The initial preparation of a disk to receive data is called "formatting." The process is also referred to as "initialization." Formatting is a process for organizing the surface of a disk into specific areas so that information can be written to those areas, and so that whatever is written can be kept track of. There are two processes involved in formatting a disk. The first is called the "low-level format." The second is called the logical (or "high-level") format. On a floppy disk (when you use the DOS FORMAT command), both procedures are carried out at the same time. For this reason, when you format a floppy disk, all data on it is erased and, for all practical purposes, cannot be recovered.

On hard disks, however, the process requires two separate steps, low-level formatting and high-level formatting. Data "lost" by accidentally re-formatting a hard disk (using DOS FORMAT) can be recovered because it is not actually erased. This assumes, however, that you have not subsequently saved data to the disk and written over the old data. A low-level re-format can be done without destroying data on the hard disk, but this takes a bit of knowledge, and should not be attempted in a casual manner.

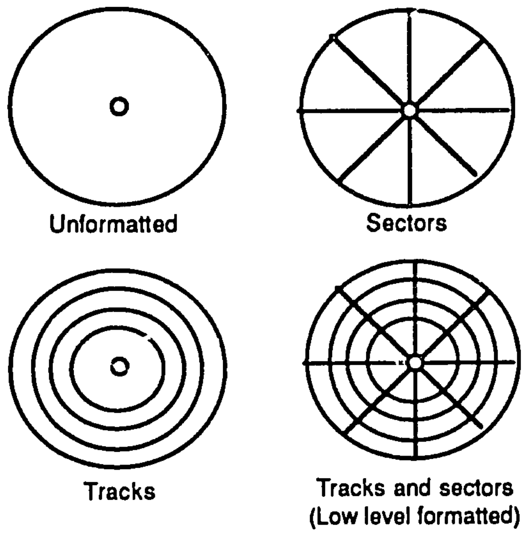
Low-level Formatting

The low-level format procedure is similar to setting up rows of chairs in a theater (with aisles running across the rows) and then numbering each aisle, row and seat. In this way the basic, physical organization of the theater is established – there are a certain number of aisles and rows, and within each row there are a certain number of chairs. The patrons who (temporarily) occupy the chairs correspond to data. The organization of the theater is thus fixed – patrons who come into the theater are not allowed to rearrange the aisles, rows or chairs, just as in writing data to a disk, you are not allowed to change the basic organization (low-level format) of the disk.

In a similar way, a disk is organized into circular areas called tracks, and the tracks are divided into sectors, as shown in Figure 6-9. This is done by writing the organization (tracks and sectors) onto the disk magnetically, much as data is written to the disk later on. During low-level formatting, defective areas on the disk surface are marked, so that they cannot be used for data later on.

Low-level formatting of a hard disk, also called "hard formatting" or "machine formatting," is normally done by the disk vendor before it is delivered to you. Therefore, it may appear that low-level formatting need not be of concern to the average user or system maintainer. But, magnetism diminishes of its own accord over time. Low-level formatting, like writing data to the disk, is done by magnetic charges on the disk surface. These charges do not last forever. Also, through normal wear or through malfunctions, the heads that write data to the disk may become misaligned and write over part of the low-level format information on the disk. Many times, problems with a hard disk can be eliminated by simply redoing the low-level format. A disk can be reformatted at the low-level without destroying data stored on the disk, but in all cases, the disk should be backed up first.

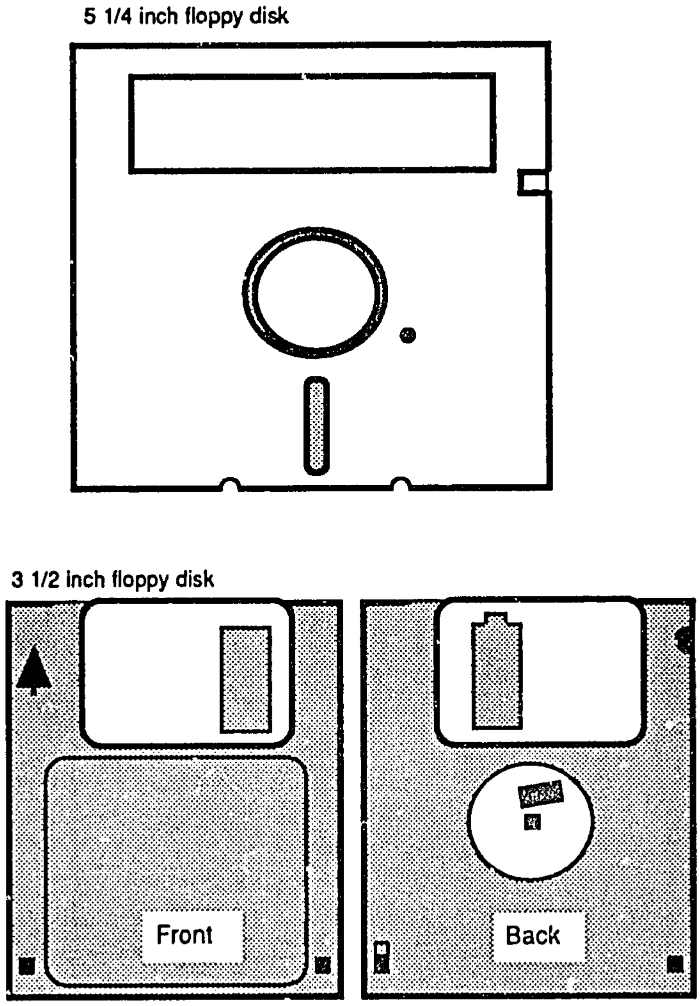
Fig. 6-9. Tracks and Sectors of a Disk Surface



Floppy Disk (Diskette) Structure

A floppy disk (or "diskette" to distinguish it from a disk within a hard drive), of course, has two sides, and both sides are used for storing data. Therefore, a floppy disk requires two read/write heads – one for the upper surface and one for the lower surface. Figure 6-10 shows a 5 1/4 inch floppy disk and a 3 1/2 inch floppy disk. Even though the 5 1/4 inch disk has a paper jacket and the 3 1/2 inch disk has a rigid plastic jacket, both are "floppies," and both operate the same way. They are mounted on a drive spindle that protrudes through the hole in the center of the disk, and they are read through a small opening (window) in the jacket, as shown.

Fig. 6-10. 5 1/4 inch and 3 1/2 inch Floppy Disks



Hard Drive Structure

In contrast to a floppy disk, a hard drive is made up of several disks. The disks are permanently affixed to the drive spindle, as illustrated in Figure 6-11. Because the disks in a hard drive are relatively rigid, they are referred to as "platters" rather than disks. For each platter, both the upper surface and the lower surface is used for storing data. And so, each platter requires two heads. As with the floppy disks, each platter is (low-level) formatted into tracks and sectors. Tracks are numbered from the outer edge of the platter toward the center, beginning with track 0. Each platter thus has a "track 0" on its upper surface and a "track 0" on its lower surface, and similarly for tracks 1,2,3 and so on.

Assume for the moment that the drive contains four platters, stacked one above the other (as they always are). There would be a total of eight "track 0s" on the drive, eight "track 1s," eight "track 2s," and so on. Notice, as also shown in Figure 6-11, that all tracks having the same number are positioned directly above one another. That is, they are vertically aligned. If you drove a nail straight down through track 0 on the top platter, it would go through each of the seven remaining track 0s beneath it. All of the track 0s, taken together, are referred to as "cylinder 0." All of the track 1s collectively form cylinder 1, and so on for the rest of the tracks.

So, we have tracks (concentric circles on the disk surface), and sectors (radial lines) coming out from the center of the platter or disk, which intersect with the tracks to define spaces on the disk, just as rows and columns intersect on a spreadsheet to form data cells. A cell's location can thus be identified by specifying: (1) the platter it is on, (2) the surface of the platter, (3) the track, and (4) the sector. What's important to know is that each sector (cell) can hold 512 bytes of data. This is the standard, at least for IBM machines.

Clusters and File Allocation Units

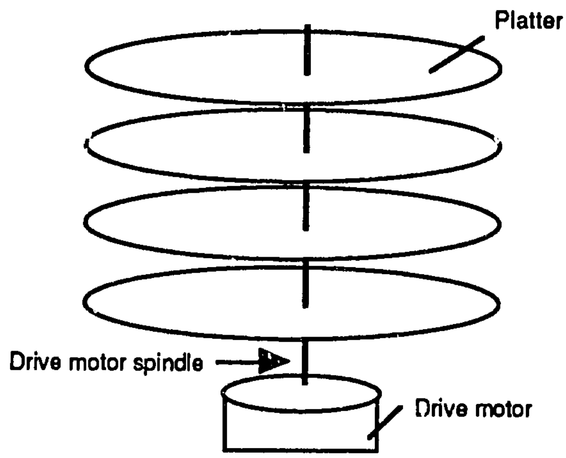
When an operating system (e.g., DOS) maps out the disk surface during the high-level format discussed later, it determines the minimum amount of space that will be allocated for writing any data to the disk. That minimum may be just one sector (512 bytes), or it may be two sectors (1024 bytes) or even four sectors (2048 bytes). For the moment, the details are not important. It is significant to note only that the minimum space allocated is called a "cluster." A cluster may be defined to be just one sector, or it may consist of more than one sector. Of late, the term cluster is being replaced with "file allocation unit." They are synonymous terms.

For any file saved to a disk, the least amount of space that will be assigned to that file is one file allocation unit (one cluster). So, even if your word processing document, for example, was nothing more than a blank page with just a single character (one byte) typed on it, it would use (i e., have allocated to it) no less than 512 bytes of space on the disk. If you then edited your document and added another 512 characters to it (a total of 513 bytes worth of characters), the first sector would be filled up with the first 512 bytes of characters and the last character (byte) would be stored in another sector of 512 bytes.

181

Fig. 6-11. Hard Drive Platters

Hard drive disks (platters) are permanently affixed to the drive motor spindle



Thus, your file would now be allocated 1024 bytes (1Kb) of disk space (two complete sectors), even though the document itself was only 513 bytes long. If your document never became any larger, the remaining 511 bytes of disk space in the second sector would be wasted – nothing from a different file could be written in that space. The unused space is called "slack."

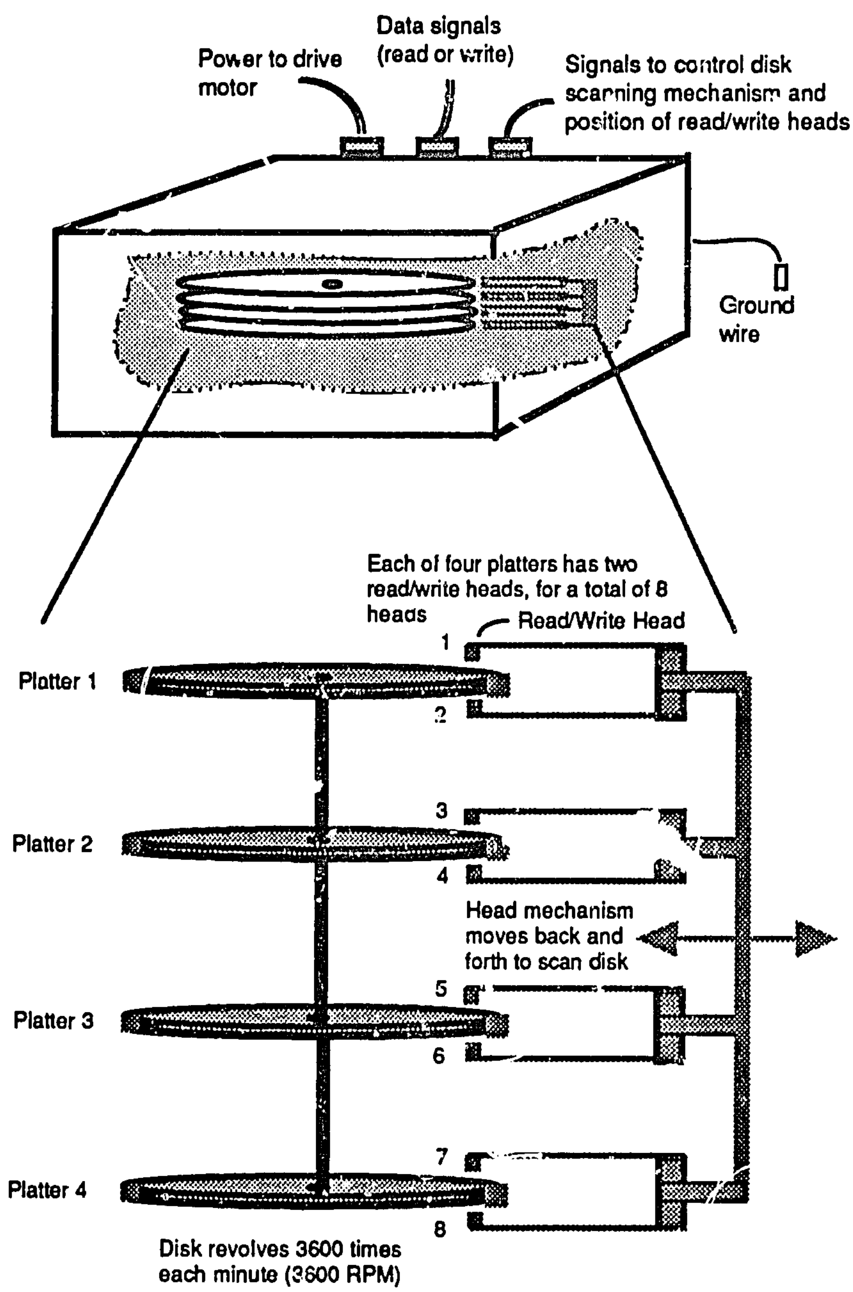
So far, our discussion of formatting has dealt with just one surface of an individual disk. A floppy disk, of course, has two surfaces, but a hard drive can contain several disks and thus proportionately more surfaces to deal with. Let us digress for just a moment, then, to look at how the disks of a hard drive are arranged.

Hard Drive Disk (Platters) and Read/Write Heads

In Figure 6-12 we see in general how the hard drive platters and read/write heads are set up. Each platter has two heads (upper and lower). The heads are attached to an arm, and the arms form a single unit that moves all of the heads across the platters in unison. That is to say, when head number 1 of platter number one is sitting on track 0, all of the other heads on the drive are also sitting on track 0 of their respective platters. All of the heads move in and out together, as a unit.

To read or write data, the heads are positioned as follows. A request to read or write data is sent from DOS via the BIOS to the hard disk controller. The controller sends a signal to the drive's stepping motor. The stepping motor moves a

Fig. 6-12. Hard Drive Platters and Read/Write Heads



specified amount in the required direction to position the heads on the needed cylinder (set of vertically aligned tracks). Incidentally, stepping motors are "detented" or ratcheted devices. Thus, as they move across the platter, stopping here and there, they make that clicking and buzzing noise that you may have been wondering about.

When the drive head gets into position, it must then wait for the needed sector to spin past it. Considering that a hard disk spins at a rate of 60 revolutions per second, it may seem that the head would not have long to wait before the desired sector arrived. But by the same token, it takes the drive time to read the data, check it, and send it back to the BIOS and DOS. The BIOS and DOS do some checking of their own, before requesting that the next sector be read. By that time, the next sector needed may have just passed by, and the head must now wait until it comes around again – another full revolution of the disk.

If the data needed were written on 10 adjacent (consecutive) sectors, it might take 10 revolutions of the disk to read all 10 sectors, even though the sectors were right next to each other on the disk. By today's standards, this would be a very slow and inefficient process indeed. The business of setting up the drive in order to optimize this interaction between disk revolution and sector reading and writing is called "interleaving," as explained below.

Interleaving

Interleaving is the numbering of disk sectors in relation to the speed at which the disk drive and the rest of the system involved can transfer data to and from the disk. Simply put, many disk drives can spin (and thus present data to the read/write heads) faster than the system (DOS and BIOS) can accept the data. The solution is to number the sectors on the disk in an order that best accommodates the speed of the system to the speed of the disk.

By way of a simple analogy, consider the following. Assume that the disk is represented by a railroad train made up of 9 cars, running around a circular track. Information is printed on the side of each car of the train. Each car represents a sector on the disk. You are the read head, and you can see the train by looking out of a small window. You can see only one car at a time. Your job is to read the information from the side of each car and relay the information to someone else. You must read what is written on every car, and you must read the cars in the order they are numbered – 1,2,3 and so on through 9.

184

Further, say each car is numbered in order, from 1 to 9. (Hard disks have a greater number of sectors, but 9 will suffice for our example.) As you watch the window, car number 1 comes into view. You read its information, and then momentarily turn your attention to relaying the information to someone else. Meanwhile, car number two passes by the window. Since you must read car number 2 next, you must wait for the train to go all the way around the track again, for car number 2 to again come into view. Under these conditions, the train would have to make 9 trips around the track in order for all 9 sectors to be read. This, of course, would waste considerable time.

Let us now assume that you can read and relay the information from one car in the time that it takes one more car to pass by. Under the conditions stated above, when you were ready to read car number 2, car number 3 would be visible at the window. Ideally, car number 2 should be in the window at that time. The solution is simply to number the cars such that their appearance in the window corresponds to the speed at which you can read and relay the information from them. In the present example, then, the cars would be numbered 1,6,2,7,3,8,4,9,5.

The process of numbering the cars (sectors) in this manner is called "interleaving," because the sector numbers are interspersed or interleaved to form a sequence other than a simple arithmetic order. The number of sectors physically present between the consecutive numbers plus 1 is called the "interleave factor," or "interleave ratio." In the above example, we have one interleaved sector. That is, there is one sector (sector 6) between sectors 1 and 2; one sector (sector 7) between sectors 2 and 3, and so on. Therefore, the interleave factor here would be 1+1 = 2, or "2:1."

High-level Formatting

Once the low-level format is completed, the hard disk still is not ready for storing data. High-level formatting is required. This is done using one or more utility programs provided as part of the operating system. An operating system, such as DOS (see Chapter 7) is software (one or more computer programs) designed specifically to manage the flow of information to and from peripheral devices such as disks, printers and monitors, and to further organize disk space for storing data, keeping track of where the data is, and creating and managing information about the data files – file (e.g., document or spreadsheet) name, size, date created, and so on.

Therefore, since the operating system has all of this responsibility, it must have something to say about how the disk is organized (formatted) for storing and retrieving information. There are many different operating systems, of which DOS is but one. Each operating system has its own way of doing things. Normally, an

183

operating system is distributed on floppy disks and then transferred (i.e., copied) to the hard disk, where it remains for normal use.

Partitions

It is possible to place more than one operating system (e.g., DOS, Unix, Xenix) on your hard disk and to use one or another of them whenever you wish. Since this manual is intended for persons who will be using DOS and who most likely will not have an operating system other than DOS on their machine, we will not elaborate here on the use of multiple operating systems. However, in formatting the hard disk for a DOS-based PC, the first thing DOS needs to know is whether there will in fact be any other operating systems on the disk. If there are, it will be necessary to create a "partition" on the disk for each operating system (to keep the operating systems separate and to ensure that no more than one such system functions at a time). Given multiple partitions, you would then have to specify which partition was to be the "active" partition at a given time.

The first step in high-level formatting is to resolve the question of partitions. Creating partitions and setting the active partition is done using the DOS program (utility) called FDISK.COM. Assuming that DOS is the only operating system involved, the entire first hard disk may be designated as a DOS partition, using the menu selections and prompts provided by the FDISK program. If this is done, the entire drive will be designated as "C:".

Note that as a convention, DOS uses the colon (:) to indicate a device. Thus, the letter indicating a disk drive is normally accompanied by a colon – A:, B:, C:, D:, etc. When entering DOS commands involving a disk drive, the colon must always be included. Otherwise, a drive is referred to simply by its letter designation, without a colon.

Assuming that you have a fairly large disk drive, say 80Mb or so, you may prefer not to have the entire drive designated as C. You may wish to "pretend" that you have several additional hard drives, for example, labeled D, E, and F. With this arrangement, you might elect to store all of your word processing programs and documents on C, your spreadsheets on D, your database files on E and, perhaps, your program management files on F. This is simply a matter of preference. You could organize your files just as well using DOS directories (covered in the next chapter), and will do so within each drive in any event.

If you have but a single physical drive, DOS allows you create these additional drives essentially through a process of bookkeeping. Space on the physical drive is allocated and assigned a letter name (D, E, etc.). You then use these drives just as if they were real, physical drives. Since they are not physical drives but rather are more like tenants on your physical drive, they are called "logical drives." They exist by virtue of the "logic" applied in creating and naming them, and in the "logic" your system employs to recognize and use them.

These logical drives are "real" drives in the sense that data written to them is physically stored on your hard drive. When you save data to logical drive D, E or F, for example, it is saved on the magnetic surfaces of your hard disk.

RAM Drives

The logical drives mentioned above should not be confused with a RAM drive (also called a "virtual" drive). A RAM drive is not a true disk drive -- it cannot permanently store data. When the PC power is turned off, the RAM drive vanishes and so does any data within it. A RAM drive is just a simulated drive. It is created by taking some of the random access memory (RAM) in your PC, giving it a name such as "D," and making that memory behave as though it were a drive. Since a RAM drive exists in the form of electronic circuitry and has no moving parts, it operates much faster than a physical drive does. This may be useful when your work requires that files be moved to and from a disk very quickly and frequently during a given work session. At the start of the session, you would simply copy the needed material into the RAM drive and use it from there. The disadvantage, however, is that if the power fails or if you forget to save all of the data in the RAM drive by copying it to a real (physical) drive, your data will be lost. Unlike a hard drive, there is no way to recover data from a RAM drive that has been erased or lost when the power is turned off.

FDISK.COM, then, is used to set and activate partitions on your hard drive, and should you choose to do so, to set up some logical drives. At this point you are ready to do what most people think of when they hear the word "format" – and that is to format the hard disk (or a floppy disk) by using the DOS program (utility) called FORMAT.COM. This is the second and final step in formatting a hard drive. It is the high-level formatting, also called "logical formatting" or "soft formatting."

Using FORMAT.COM

For most of us, formatting is encountered most often with floppy disks (or "diskettes"), so we will deal with "floppys" first. When a floppy disk is purchased, it is not ready to receive data. Like the hard disk, it requires both low-level and high-level formatting. Unlike the hard disk, however, both low and high-level formatting on the floppy disk are done by FORMAT.COM in a single step. Note that although 3 1/2 inch floppy disks are enclosed in a rigid plastic case and are not as flexible as the old 5 1/4 inch floppys, the 3 1/2 inch disks are not a form of hard disk. Inside that nice plastic case, they are just as "floppy" as they ever were.

Assuming that we have our DOS utilities (programs) residing on our C drive (hard drive) and we wish to format a floppy disk in our A drive, we would insert the floppy into the A drive and type the following command at the DOS prompt (use upper case letters or lower case letters – it makes no difference to DOS):

C>FORMAT A:

The command typed at the DOS prompt is simply the name of the utility program FORMAT.COM, with the suffix ".COM" omitted.

DOS will then automatically format (or attempt to format) the floppy disk to the full capacity provided by the floppy drive. Note the distinction made here between the disk and the drive. Since floppy drives and floppy disks exist in a variety of capacities, and since a disk of one capacity will often fit into a drive of a different capacity, problems with formatting can arise.

DOS will also create a File Allocation Table (FAT) and a root directory on the disk (explained later). The symbol denoting the root directory is the backslash (\).

Assuming for the moment that the disk and drive in question are properly matched, the FORMAT command has only two variations of interest here. One is the option to format the floppy such that it can be used to boot up the PC. The other allows you to name the disk (actually, the root directory), i.e., to create a "volume name."

To format the floppy so that it can be used to boot up the PC, the characters "/s" (a backslash and an "s") are appended to the command as shown below. A disk formatted in this manner is called a "bootable" disk.

C>FORMAT A:/s

This will format the disk and meanwhile will transfer the two PC DOS system files, IBMBIO.COM and IBMDOS.COM, and COMMAND.COM (or their MS-DOS equivalents) to the disk. The first two files are "hidden" – their names will not appear when you use the DOS directory command (DIR) to see what files are on the disk.

To assign a volume name to the floppy's root directory during formatting, enter:

C>FORMAT A:/v

When the disk is formatted you will be prompted to enter the name you wish to give to the disk (volume). Enter the name (up to 11 characters) and press ENTER. For illustration, we'll use the name GOODSTUFF.

After this is done, you can see the volume name by typing VOL A: at the DOS prompt:

C:>VOL A:

The response will be:

Volume in drive A is GOODSTUFF

You can use both options at the same time. Just type:

C:>FORMAT A:/s/v

This will produce a formatted, bootable disk having the volume name of your choice.

To format the hard drive, you will need a boot disk (one that has on it the DOS hidden system files), plus a copy of COMMAND.COM and a copy of FORMAT.COM. Boot the system from the (A:) floppy drive. The prompt A> will appear on your display. When it does, type:

FORMAT c:/s

This will format the hard disk, and will meanwhile transfer the hidden system files and COMMAND.COM to the hard drive. It will also create a File Allocation Table (FAT) and a root directory on the disk.

When the PC boots up, it looks for a copy of COMMAND.COM in the root directory. If COMMAND.COM is not found, the following error message is displayed:

"Missing command interpreter"

Unless COMMAND.COM is available to the PC, the PC will not operate.

Now remove the floppy disk from drive A. Turn off the computer, wait about 10 seconds, and then turn it on again. If all has been done correctly, the machine will boot up from the C drive and the DOS prompt, C>, will appear on your display.

File Allocation Table (FAT)

The File Allocation Table (FAT) is a table created on a disk by the operating system, to keep track of where files are stored on the disk. We have seen that

181

low-level formatting divides the surface of a disk into sectors and assigns a number to each sector. In all DOS formats, a sector will hold 512 bytes of information. When DOS writes a file (e.g., data or a word processing document) to a disk, DOS first divides the file into "chunks" of 512 bytes each. DOS then writes the chunks, one by one, into sectors on the disk that are available for use ("free" sectors). Free sectors are those that are not already in use by other data (files), or that have not been marked as defective during the format process, or that have not been reserved by DOS for other uses.

The system for keeping track of files is something like a treasure hunt – you start off from a certain place with an address to go to. When you get there, you find a message telling you where to go next. You go to the next place and find another message telling you where to go next. This continues until you find a message that says the game is over. The root directory and the File Allocation Table work together in this way, as explained below.

The root directory stores the first disk location (cluster) at which data from a file is stored. With that information you could go to the disk and find the first part of the file, but you wouldn't know where to go next. You don't have the rest of the treasure map. The information you need is stored in the FAT. The FAT contains the address of each cluster on the disk. Next to that address is shown the place (sector) where you will find the next piece of the file.

For example, as shown in Figure 6-13, the root directory may say that the first part of file "ABC" is stored in cluster 100 (starting cluster = 100). You go to the FAT table (actually, DOS does all of the work for you) and find the table cell for cluster 100. In that cell, it tells you where to find the next piece of file ABC, for example in cluster 102. So now you could go out to the disk, find cluster 102, and get that piece of the file.

Back to the FAT table again, we go to cell 102 and there it tells us that the next piece of the file is located in cluster 108. This process is repeated until we find an entry in the FAT that says we have reached the end of the file. In this example, the final piece of file ABC is located in cluster 109. The FAT entry for cell 109 is a special number that indicates the file is ended.

Chains and Fragmentation

Our file, ABC, as mentioned above, exists on the disk as a "chain" of clusters. Notice that in our example, the clusters are not contiguous – they do not fall in sequence one after the other, nor do they have to be in any particular numerical order of clusters. When you save a file to a disk, DOS looks for the next available ("free") cluster and writes your data to it. As files are edited, expanded, shortened and erased, space become available here and there on the disk, and DOS uses it accordingly. Files that are stored in clusters scattered around the disk are said to be "fragmented."

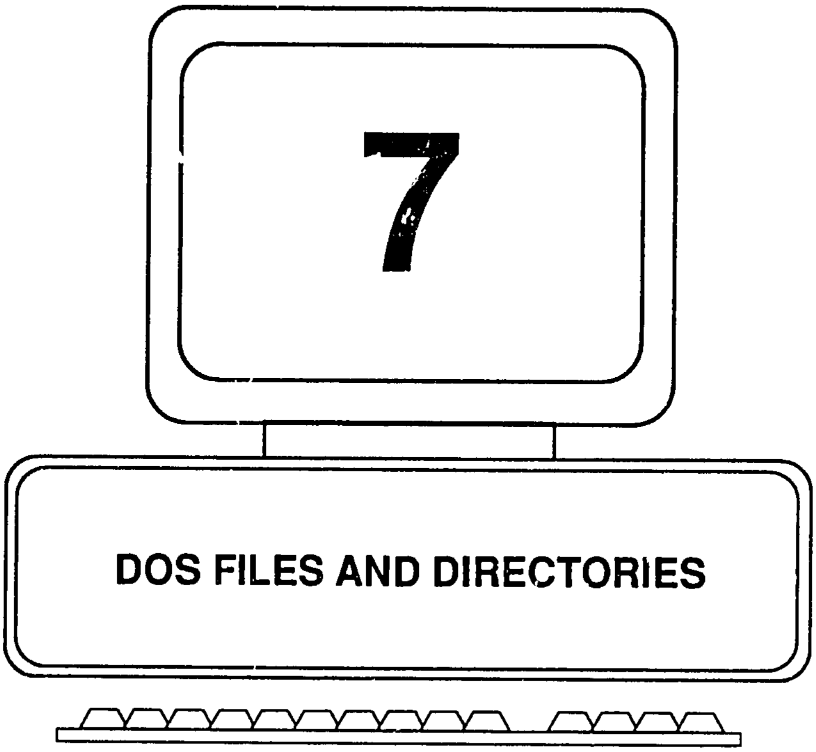
As the number and size of these fragmented files grows, and as your hard disk becomes full, it takes DOS longer to hunt around looking for each cluster of a file. The result is that your hard disk appears to be slowing down. In fact, it is not operating any slower than normal— it is just doing more work, which requires more time, to find the clusters for your files, and to find space available to write new files to the disk. As you can imagine, any damage to the FAT would cause serious problems in accessing your files on the disk. For this reason, DOS always maintains a duplicate copy of the FAT. If the first FAT fails, the duplicate is not automatically used by DOS, but it can be used by any of several file recovery programs (such as the Norton Utilities) to reconstruct a primary FAT that DOS can use.

Fig. 6-13. Relationship of Root Directory to FAT Table

Root Directory
File Name: ABC Date & Time Created File Size Starting Cluster = 100
File Name: XYZ Date & Time Created File Size Starting Cluster = 104
ETC.

	FAT TABLE (CONCEPTUAL)
98	0000
99	0000
100	102
101	103
102	108
104	104
105	FFF7
106	FFF7
107	Special number here indicates end of file
108	109
109	Special number here indicates end of file

192


7
DOS FILES AND DIRECTORIES

CHAPTER 7: DOS FILES AND DIRECTORIES

DOS Files and Directories

The information stored on a disk falls into two main categories. One category concerns the information that organizes the disk into areas (tracks, cylinders, sectors) for storing your files, and for keeping track of where the files are physically located on the disk. The other category concerns the files (your programs, documents and data) that you store (save) on the disk. Since we will be talking extensively about files, it is important to understand at the outset the general definition of the word "file," and the different kinds of files we will meet later on. We will then turn to creating and using DOS directories.

In common English usage, the word "file" often refers to a collection of papers or documents. When used in relation to computers, however, the word file means one specific program, document, spreadsheet or graphic, or set of data stored under a single name.

Each document you create on the word processor is a file, regardless of how few or how many pages it may have. Each spreadsheet you create is a file. Each database is a file. Although you may create several database files and, when using them, link them together to form a larger data base, each individual file always retains its own name and identity. And, each file is stored separately on the disk, regardless of how it may be used in relation to other files.

A file must always have a name. Without a name it cannot be saved or retrieved. So, when you see names such as COMMAND.COM, CONFIG.SYS, AUTOEXEC.BAT, ANSI.SYS, REPORT.WK1, each one is a file. If it has a name, it is a file. Files are of two general kinds – executable files and data files.

Executable Files

Executable files come in a variety of forms. Among them are program files, device drivers and batch files.

Program Files

An executable file consists of instructions (a "program") that the computer can perform (execute). When you purchase a word processing "package" such as Word Perfect, or a spreadsheet package such as Lotus 1-2-3, each package provides one or more floppy disks containing numerous executable files (along with other kinds). As you do word processing or create spreadsheets, the computer calls upon these executable files in order to accept input (text, numbers) and to carry out the commands you enter at the keyboard. A collection of executable files designed for a specific purpose such as word processing or creating spreadsheets or creating database files is called an "application."

When you "start up" a work session, say for word processing, you first activate the main program (executable file) by typing its name. The file (program) is then loaded into your computer memory (into the temporary, electronic workspace called "random access memory," or RAM). From there on you interact with that file (program) to create, edit, save and retrieve documents. Meanwhile, the main word processor program calls (uses) other of its own files to help it do its work.

The names of executable files usually (but not always) end with ".exe" or ".com". Those that are part of an application (or a game) will run when you type their name at the DOS prompt. Others are run not by you, but by the PC. One example is the COMMAND.COM file. If you type "COMMAND" at the DOS prompt, nothing happens except that the screen displays the version of DOS that is on your machine. Your PC itself uses COMMAND.COM. This file is essential to running your PC under the DOS operating system. Its job is to translate (interpret) the DOS commands that you enter at the keyboard into the language that the PC understands. If DOS cannot find COMMAND.COM on your system, it will inform you:

"Bad or missing command interpreter."

Another kind of executable file is the "system" file. A system file is used by the computer to manage the inner workings of the machine, to interact with the basic input-output system (BIOS) that is built into your PC, and to communicate with the various peripheral devices (keyboard, display monitor, printer, etc.) connected to the PC.

Device Drivers

Certain executable files that apply to peripheral equipment such as the keyboard, mouse, printer and other, similar items are called "drivers." This type of file is used by your computer to set up (configure) the system so it can communicate with these items of equipment. For this reason, these files are called "device drivers." You do not use these driver files directly (e.g., as you would use a spreadsheet program or word processor program). Instead, when you turn on your PC or use a particular application (word processing, etc.), the PC looks for these driver files and uses them without any commands from you.

Batch Files

Finally, there is another type of executable file – the "batch" file. The names of these files must end with ".BAT". Otherwise, DOS will not run (execute) them. A batch file is nothing more than a set, series, string or collection (i.e., a "batch" as in "several") of DOS commands that you would normally enter at the keyboard. As explained later in this chapter, you can save a series of DOS commands in a single file – a batch file. Then, all you need do is type in the name of the batch file, and all of the commands in it will be executed. Mainly, batch files are used to reduce the laborious, repetitive typing of DOS commands.

Batch Files and Others in ASCII Format

You can create (write) and save a batch file working from the DOS prompt. Or, you can write it on a word processor, provided you save it in ASCII format. Most word processing applications provide an option (menu selection) for saving a file as an ASCII file. ASCII (pronounced "ass-key") stands for "American Standard Code for Information Interchange."

During the evolution of computers, every developer had a different way of representing alphabetic and numeric characters (which are stored – coded – as numbers in the machine). This made it difficult to transfer files from one system to another. So industry representatives got together and agreed on a standard method for coding the characters. The result was the ASCII format. Today, just about every application (word processor, spreadsheet, database) still has its own way of coding characters. But, each one allows you to create an "ASCII" version of your spreadsheet or document. That version can then be read by a different application, since they all share an understanding of the ASCII format (character coding system).

DOS can read and display ASCII files directly. To display an ASCII file on your monitor, simply type the command TYPE at the DOS prompt, followed by a space and then the name of the ASCII file.

For example, when you purchase a new application, you will often find on one of its floppy disks a file named READ.ME. Usually, the file contains important information that arrived too late to be printed in the application manual. To see the file, type the following command at the DOS prompt:

TYPE READ.ME (yes, you do type the word TYPE)

If the file is on your A drive, the screen would look like this:

A>TYPE READ.ME

If the file is larger than one screenful, it will scroll past faster than you can read it. To stop the scrolling, hold down the CTRL key and then press S. Careful – if you press the S before holding down the CTRL key, the scrolling will not stop. To start scrolling again, press any letter key or the space bar.

You would use this same method to read any batch file or your CONFIG.SYS file.. Reading a file in this way will not alter the file.

The AUTOEXEC.BAT File

As we have just seen, there is nothing complicated about a batch file. The AUTOEXEC.BAT file is no exception. The only difference between this batch file and any other batch file is that it has a special name. When your PC boots up, it goes through an extensive routine of testing and checking, loading drivers, and other activities needed to get things ready for the day's work. During this process, DOS looks for a file named AUTOEXEC.BAT. If DOS finds the file, it executes whatever commands are in it. Since DOS will AUTOmatically EXECute this BATch file, it bears the sensible name of AUTOEXEC.BAT.

The AUTOEXEC.BAT file normally contains a series of DOS commands that you would otherwise have to key in manually each time you started up your PC. For example, suppose you want the prompt always to show the directory you are working in at the moment (the "current" directory), rather than just "C>," regardless of the directory you are using. To have this happen, the following command would be placed in the AUTOEXEC.BAT file:

prompt pg

Another and very important command found in the AUTOEXEC.BAT file is the PATH command. The PATH tells DOS where to look for a file if it can't find that file in the directory you are working in at the moment. The PATH is in fact just a string of directory names, each separated by a semi-colon (;). DOS will search each directory in the string until it finds the executable file it is looking for. Note that data files cannot be found in this manner, only executable files.

Recall from our discussion earlier, that whenever you type the name of an executable file at the DOS prompt, DOS will run (execute) the file. That is, DOS will run it if it can find it. As we shall see later, your hard disk should be organized into directories (essentially these are named storage spaces), and files are stored in the directories according to their type or use.

Whenever you are using your PC, you are "working from" one or another of these directories. DOS know where you are, and when you ask DOS to run a file, DOS assumes you mean for it to run a file that is in the directory you are in at the moment. This directory is also called the "current" directory or the "default" directory. If the file is not in the current directory, and unless you have given DOS a path (a string of directories) by which to extend its to search, it will give up and report: "File not found."

To keep things simple for the moment, we'll assume that you have five directories in all. These are the root directory, C:\ (which everyone must have), plus directories ABC, DEF, GHI and JKL. In real life, your directories would have more sensible names, but these latter names will do for now. We'll assume further that when you command DOS to run a file, you expect DOS to search all of these directories in order to find it. So, your path command would be this:

path C:\;C:\ABC;C:\DEF;C:\GHI;C:\JKL

Obviously you would not care to type this command every time you started up your PC. So, the easiest thing to do is to put it into your AUTOEXEC.BAT file. Thus far, our AUTOEXEC.BAT file would have two commands in it:

prompt pg

path C:\;C:\ABC;C:\DEF;C:\GHI;C:\JKL

DOS will search through these directories in the order they appear in the path command. So it is a good idea to put the directories in the order they will be most frequently searched.

Using AUTOEXEC.BAT to Bring Up an Application

To close our discussion on the AUTOEXEC.BAT file, let's assume that you are going to start work every day using your spreadsheet program (e.g., Lotus 1-2-3). You would like to have your PC boot up and then automatically bring up 1-2-3.

We'll assume that your spreadsheet application files are stored in a directory named LOTUS which exists immediately below the root directory, C:\.

The main executable file (program) for Lotus 1-2-3 is named 123.EXE. To run it, you would normally first have to change from the root directory (where your PC leaves you after it boots up) to directory LOTUS, and then type the file name, 123, to run the file.

The DOS commands you would have to type manually are:

CD\LOTUS (change directory to LOTUS)

123 (run file 123.EXE)

We now add these commands to our AUTOEXEC.BAT file:

prompt pg (set prompt to show current directory)

path C:\;C:\ABC;C:\DEF;C:\GHI;C:\JKL (search these directories)

CD\LOTUS (change current directory to LOTUS)

123 (run file 123.EXE)

Now when the PC boots up, it will automatically set the prompt to show the current directory, set the path, change to the LOTUS directory and, finally, run (bring up) 1-2-3. Alternatively, if you added the directory C:\LOTUS to the path command as follows:

C:\;C:\ABC;C:\DEF;C:\GHI;C:\JKL;C:\LOTUS

you would not need to place the CD\LOTUS command in the AUTOEXEC.BAT file. Just the command 123 would suffice, because 123 is an executable file, and so DOS would find it by using the path, and then run it.

Where to Keep the AUTOEXEC.BAT File

If you have an AUTOEXEC.BAT file, it must be kept in the root directory in order for DOS to use it during bootup. But, you do not absolutely have to have an AUTOEXEC.BAT file. Most people, however, find it helpful. Further, when you troubleshoot a PC, you will be interested to know exactly what commands are in the AUTOEXEC.BAT file if there is one. The file is commonly used to automatically load "pop up" utility programs that can be accessed (used) during a work session, and other "helpful" programs such as "screen blanking," "schedulers," "rolodexes," and such.

Once they are loaded, these kinds of files (like device driver files loaded from the CONFIG.SYS file) may remain in memory (RAM) after you are done using them (have terminated them). When you terminate these files, they stay resident in your PC's memory (RAM). Thus they are called terminate-and-stay-resident (TSR) files. TSR files are sometimes the cause of problems with allocation and use of RAM when running application files such as for word processing. This can cause "memory error" messages to appear, or it may lock up ("hang") the PC in the middle of a work session on what undoubtedly will be your most important document.

Data Files

A data file does not contain instructions that the computer can execute (except for imbedded macros, printer commands and similar instructions). Rather, it contains the text document, spreadsheet, graphic, or data base that you or someone else has created. Thus, the name "data file" does not imply data in the sense of just "numeric" data. It may be a document, a drawing, a schedule, a spreadsheet – most often, it is the "product" you create when using an application.

When you save a file (write it to a disk), DOS records the file name; time and date; size of file (in bytes); and whether it is a hidden file, "read-only" file, or "regular" file. DOS breaks up your files into "chunks," and writes the chunks into whatever spaces (sectors or File Allocation Units) are available on your disk.

Your document thus exists in pieces, in different physical locations on the disk. The location of the first chunk is stored in your root directory. The location each of the remaining chunk is stored in a File Allocation Table (FAT), also on your disk. The addresses of the different locations are linked to form a chain.

When DOS retrieves a file from the disk, it goes first to the root directory to find the address of the first chunks of your file on the disk, and then it works through the file allocation table (FAT), chunk by chunk, to find the remaining pieces of your file. The filing structure or organization of the disk (its format), the addresses of the pieces of your files, and the contents of the files themselves are all written on the disk.

The files (data) are written by means of microscopically small magnetic charges on the disk surface. The disk drive, which rotates the disk and moves the writing, reading and erasing mechanisms ("heads") across the disk surface, is an electromechanical device that has many moving parts. Considering all of this, there are many ways in which the device can fail.

DOS Directories

Many people seem to be uncomfortable working with DOS subdirectories and thus either do not use them to full advantage, or else they become overly dependent on a utility program for dealing with their directories "at arm's length." Whether you create subdirectories by using the DOS "make directory" command or by using a utility program, it pays to understand something about their structure and behavior, both to manage your own files and to help others who are having problems with directories.

Directory Structure

As noted earlier, the root directory is created by the high level formatting process, but this single directory by itself is not adequate for organizing your program files (applications) and your data and document files. Trying to keep everything in just one directory would be a nightmare. So, DOS makes it easy for you to create and name other directories in which to keep your files. The root directory, however, is the "boss" directory. It has special properties and limitations that the other directories you create do not have. The other directories are properly called "subdirectories," because they exist as dependents of the root directory, much as children are shown as dependents of their parents in a family tree.

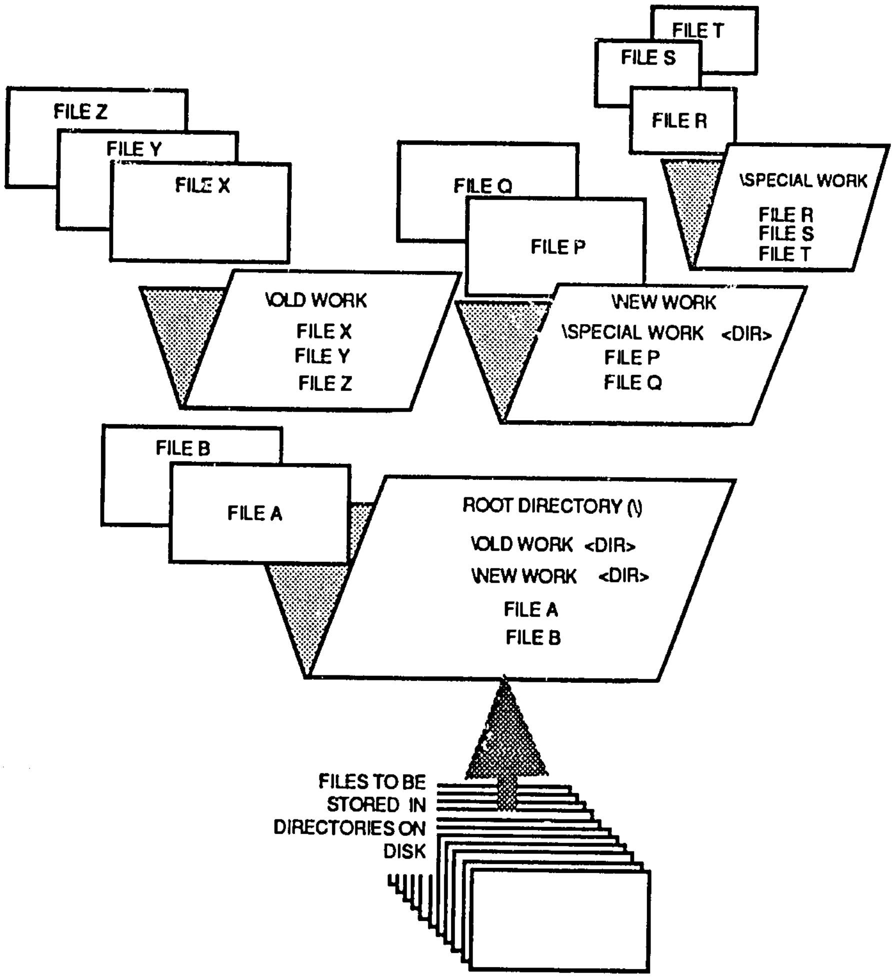
If, for the moment, we forget about the special properties of the root directory, we can say that any directory or subdirectory is basically a document like the ones you create on a word processor. The only difference is that, instead of storing text, a directory stores the names of files (along with the names of any subdirectories existing at the next level beneath it, as we shall see in a moment). The only limitation to the number of entries in a directory (other than the root directory) is the amount of space available on your disk.

Do not become confused at this point about subdirectories. They are simply a filing system. But there is an important matter of semantics here. Normally you might think of a "file" as a collection of papers or documents. In computer parlance, a "file" is a specific, individual thing to be stored on the disk. A file can be a word processing document, a computer program, a spreadsheet, a graph, and so on. On the other hand, a directory is like a paper folder that holds documents for you. A directory can be empty, just as a folder can be empty. Or a directory can hold files (individual documents, programs, graphs, etc.). A directory (folder) can also hold another folder.

Let's look at Figure 7-1 and see how all of this works. At the bottom of the figure, we have a stack of files (documents, programs, etc.) that we wish to store on our hard drive, C. Assuming that we have just received our PC and that the formatting has been done, we would find that we already have a folder set up for us. It is the root directory, and its name is C:\ (read as "C colon backslash").

We could store everything in this folder (up to the limit of the number of files permitted in the root directory) -- the applications programs (files) such as the word processing, spreadsheet, and database software products that we purchase from a vendor.

Fig. 7-1. DOS Root Directory and Subdirectories



As we use these applications to create text documents, spreadsheets and so on, we could also store all of our work in this folder (the root directory), up to the number of files permitted in the root directory. In any event, we would have so many files in our root directory that it would take forever to hunt around in there to find our work. Also, DOS does not permit two files within the same directory to have the same name. Sometimes it is necessary to have different files with the same name. Therefore, each must be stored in a different directory.

Subdirectories

For simple illustration, let's say that we wish to store only two files in our root directory. We'll call them File A and File B, as shown in Figure 7-1. And, that we create two new folders (subdirectories) called "Old Work" and "New Work." We can now store files in these folders as well. Furthermore, we can store the folders themselves inside of the main folder (root directory).

As shown in Figure 7-1, we now store three files (File X, File Y, and File Z) in the folder named "Oldwork." Similarly, we store two files, (File P and File Q) in the folder called "New Work." Notice the front of the root directory folder. It shows its own name (\), and it shows the names of the two subdirectories, New Work <DIR> and Old Work <DIR>. DOS indicates a subdirectory name by enclosing DIR in < >. And, we also see there the names of the files that are stored in the root directory (File A and File B).

Similarly, the cover of folder "Old Work" shows the names of the files contained in it. Since that folder does not itself contain any other folders (subdirectories), that's all there is to it.

But, now we find that as we do work and create more documents (files) it would be convenient to store some of these files in another folder, right there inside of the New Work Folder. So, we create yet another folder (subdirectory) and call it "Special Work." Inside the Special Work folder, we now store three files (File R, File S, and File T).

Look at the cover of the New Work Folder. It shows the name of the folder (subdirectory) "Special Work," that is stored inside of itself; and, it lists the names File P and File Q, which are stored in the New Work folder itself.

Why does the cover of the root directory folder not show that there is a subdirectory called "Special Work"? The answer is that the listing of the contents of any directory or subdirectory will show only those subdirectories that exist immediately beneath it -- like a parent who tells the names of his or her children, but fails to mention any grandchildren.

Looking once more at Figure 7-1, we see that it begins to resemble a tree, with its subdirectories branching upward, and from there branching further into the files that are its leaves. But we have spoken of "sub" directories, and we have talked about a subdirectory existing beneath its parent directory. Strictly speaking the DOS directory tree is more like a family tree, with ancestors at the top and the younger generations branching downward. So, let's turn our example tree upside down, as shown in Figure 7-2, and examine it further.

In Figure 7-2 (Diagram A), we have added lines connecting the directories according to their parentage. We see that the subdirectories Old Work and New Work descend directly from the root directory (as its children). The Special Work subdirectory is the offspring of subdirectory New Work. The Special Work subdirectory is the "grandchild" of the root directory because it exists two generations (levels) down from the root directory.

At the bottom of the page (Diagram B), we have drawn the tree structure in skeleton form. In the skeleton diagram we have done two things with the directory names. First, since DOS does not allow spaces within a file name or a directory name, we have combined the words, i.e., Old Work has become Oldwork.

Since DOS also limits us to 8 characters in a name, we have shortened Special Work to SPECWORK. Also, we have in each case placed the subdirectory's lineage (parentage) in front of its name. You can think of this as being the subdirectory's "full name" or "proper name." In the world of DOS, what we have now shown is called the "path name," or "absolute path name."

Path Names

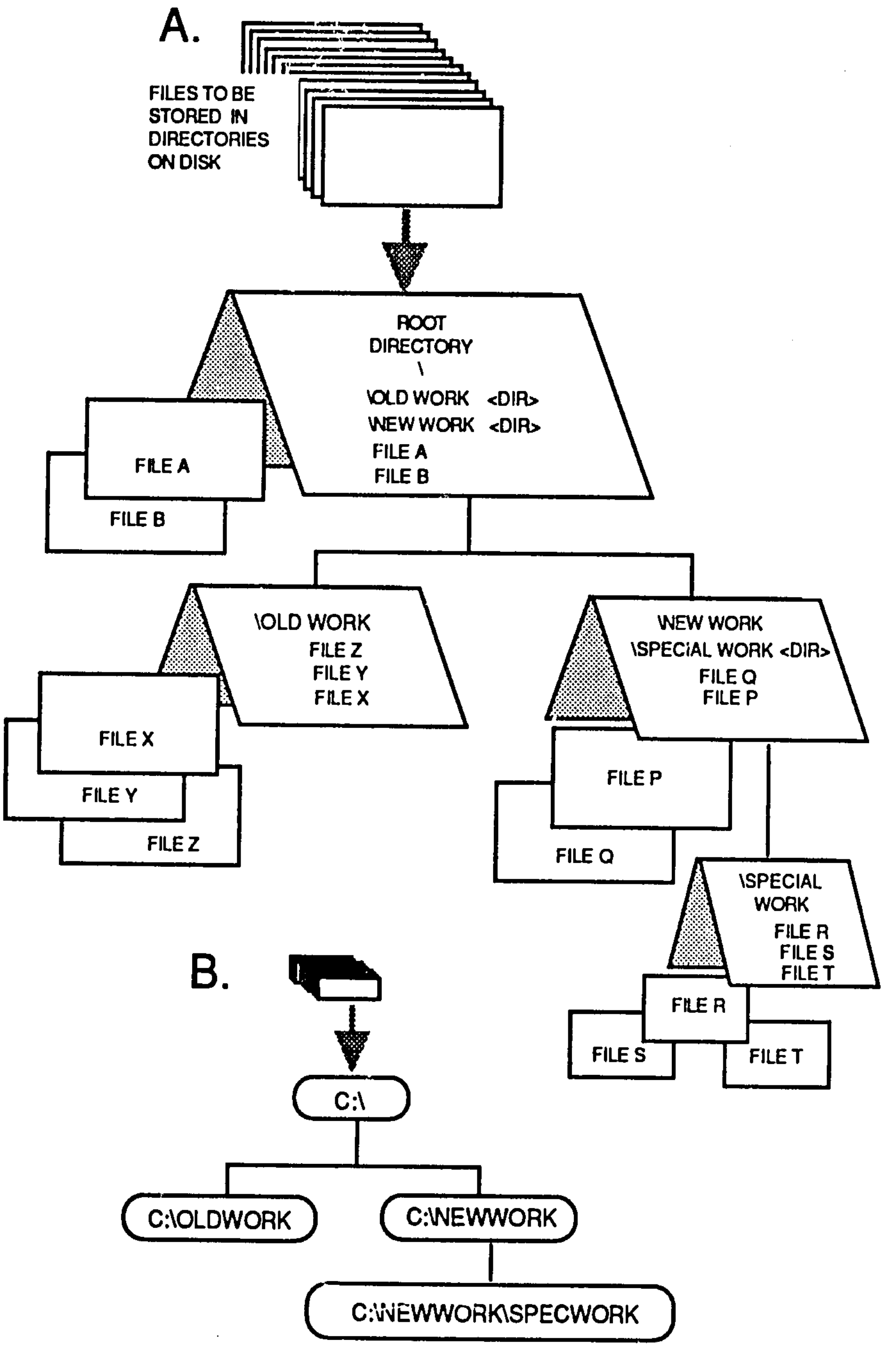
Here is where persons new to DOS often become disenchanted – nobody likes to see these long strings of letters and backlashes. But, please have a little patience. It is really quite a simple matter.

Let's examine the longest path name shown in the diagram and see what it has to do with a "path:"

C:\NEWWORK\SPECWORK

The name, as shown, is really a set of directions telling you where the subdirectory SPECWORK resides. If someone asked, "Where would I find the directory called SPECWORK, you would say, "Go to the C drive; from there go to the root directory (\); from the root directory proceed to the subdirectory called NEWWORK; and from there, go to the subdirectory called SPECWORK."

Fig. 7-2. DOS Directory Structure as an "Upside-down Tree"



When you use your computer and tell DOS to find (or copy, or erase) a data file, it is not enough to simply say, for example, "go to SPECWORK," and then do this or that operation. DOS needs a full set of directions. It needs to be told the path to follow in order to find the file. If you do not specify where the file is located, DOS will try to find it in the directory you are working in at the moment (i.e., the "current" directory). If DOS does not find the data file in the current directory it will say to you, "File not found," and give up the search. But, if the file in question is an executable file (not a data file, but rather a program that DOS can execute), DOS will look to see whether you have set up a path to search for such files.

By now it should be clear that files can reside in the root directory or in a subdirectory. Suppose someone asked, "Where would I find the file whose name is File S?" The real question is, "What is the path to FILE S?" If you think of a file (document, program, graph, etc.) as an adopted child of the directory in which it lives, the answer here is simple. We just tack the file name onto the path name of the subdirectory containing the file. Of course, we need to use another backlash ahead of the file name, otherwise everything would run together and DOS would get confused. So, the answer is:

C:\NEWWORK\SPECWORK\FILE_S

Oops! We've been cheating a little bit. We can't really name a file "FILE S," because spaces are not allowed in a file name. We would have to call it FILE-S, FILE_S, or something that used the characters allowed by DOS for a file name. Here we have used FILE_S, because the underline character is allowed in a file name. Now we can see clearly from the above answer that FILE_S resides in subdirectory SPECWORK; SPECWORK is a subdirectory of NEWWORK; and NEWWORK is a subdirectory of the root directory, \.

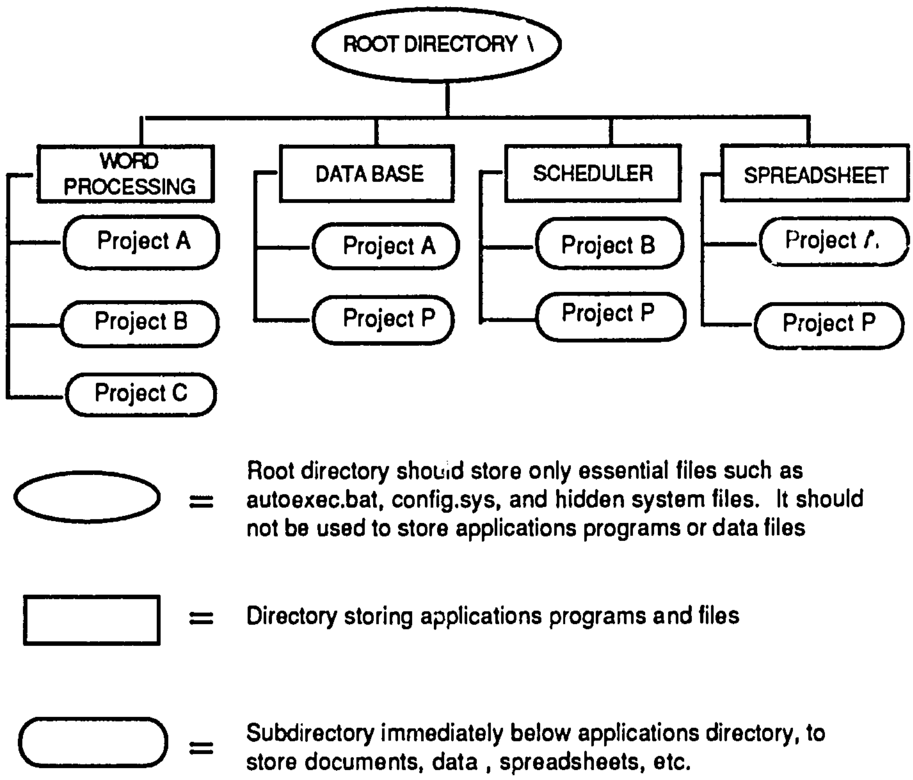
Finally, we see that in this instance, the root directory exists on drive C. As you might imagine, things can become quite complicated if you create a directory structure having too many levels. The path names would simply be too long to work with easily. That is why we said earlier that one or two levels down from the root directory is usually sufficient to organize all of the files on our hard disk.

A Typical DOS Directory Setup

Figure 7-3 shows a typical directory setup. Your own arrangement may be somewhat different from this, but the basic idea is to organize files according to functional areas or subject matter. Also, it is a good idea not to mix within a single directory the application programs (like all of those files that come with Word Perfect, Lotus 1-2-3, and DBase) with the documents or data files you create by using those applications.

186

Fig. 7-3. A Typical DOS Directory Setup



Creating and Removing Directories

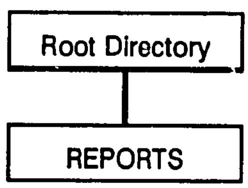
First we'll discuss what we want to do, then we'll deal with the commands for doing it. The parent directory is the root directory. Its "short name" or symbol is the backlash, \ . There is no special reason for having selected the backlash. In many cases, the use of a particular symbol is based more on the characters we have available on the keyboard than for any other reason. A directory exists on a disk drive (actually it exists as "information" written on the surface of the disk), and so it is designated first by its drive letter, and then by its symbol. The root directory of your C drive thus is named C:\. Note that we must include the colon (:) after the drive letter. A colon tells DOS that we are referring to a device – in this case the device is a disk drive.

The very first subdirectory you create must stem from the root directory. From there you have two choices. You can either create additional subdirectories from the root directory, or you can create subdirectories beneath existing subdirectories.

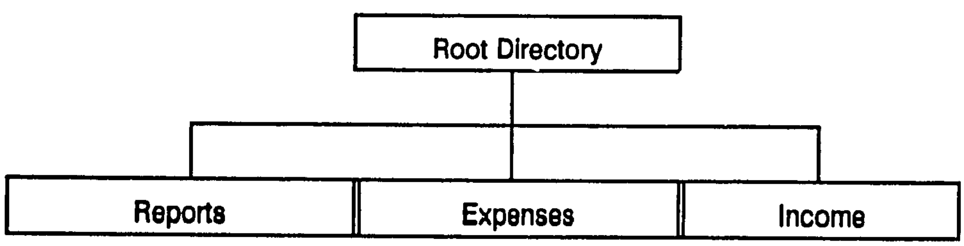
Two important things to remember:

1. Whenever you use your PC, you are always working from within one specific directory at a time. Even though you may have several directories, only one of them is active at a given moment. If you think of your directories as "rooms" within which you store things, to do any work you must be in one of those rooms at the time. The room you are in is called the "current" directory. It is also called the "default" directory because, unless you tell DOS otherwise, DOS will assume that your commands refer to the directory you are working in at the moment.

2. When you tell DOS to create a new directory, DOS creates the new directory immediately below the directory you are currently working in. For example, if you are in the root directory and you tell DOS to create a new directory called "REPORTS," your directory structure would look like this:



Now let's say you wish to have two more directories, one for expenses and one for income. And, you want them to be subdirectories below the REPORTS directory. If you simply enter the commands to make these directories as you just did for the REPORTS directory, you would not get the desired result. Instead, your directory structure would look like the following diagram.



208

This would happen because you were still working from the root directory when you created the new directories, EXPENSE and INCOME. Therefore, DOS created them immediately below the root directory.

To get the structure you need, you can "go into" the REPORTS directory and create its subdirectories from there. In other words, you change the current (active) directory from being the root directory to being the REPORTS directory. Then, when you create the directories, EXPENSES and INCOME, they will be subdirectories below the reports directory.

Alternatively, it is possible to create a directory regardless of which directory is the current directory. To do so, simply use the "full" name of the desired directory, as illustrated in the following example.

Assume the directory structure at present contains C:\REPORTS, C:\INCOME, and C:\EXPENSES as shown in the above example. Assume further that the current directory is C:\REPORTS, and you now wish to create a subdirectory called "PERSONAL" below the EXPENSE directory. The command would be:

MD C:\EXPENSES\PERSONAL [enter]

Removing Directories

Let's assume that we wish to remove the three directories we have just created. We will cover the basic approach first, and then the specific commands.

To remove a directory, you must first erase all of the files within it. Then, to remove the empty directory, you must work from the directory immediately above it in your structure. That is, to issue a command to remove a directory, the active (current) directory must be the directory immediately above the one you wish to remove.

Since we have not put any files into them, there are no files to erase. To remove either the EXPENSE directory or the INCOME directory, we must work from the directory immediately above them, which is the REPORTS directory – REPORTS must be the current directory.

We then tell DOS:

- remove the EXPENSE directory, and then
- remove the INCOME directory

The order in which you remove them does not matter.

Now, we are still "sitting in" the REPORTS directory. We cannot remove it from where we are – that would be like sawing off a limb that we were sitting on. We first must move up one level, into the root directory. From there, we tell DOS to remove the REPORTS directory.

We are now back to having just the root directory. By the way, you cannot remove the root directory, ever. To remove a directory you must work from the level above it. Since the root directory is at the top of the structure, you cannot get above it. Therefore, you cannot remove it even if you tried.

Now that we have seen what is generally involved in creating and removing directories, we will deal with the specific commands for doing these operations. But first, the question arises – How do we know what directory we are in at the moment? That is, how can we see what the current directory is?

When a PC is first set up and you turn it on, what you usually see on the screen is the DOS prompt, >, preceded by the letter of the drive you are using. So, the prompt appears as C>. Even as you change from one directory to another, the prompt remains as C> unless DOS has been told to make it otherwise.

It is generally convenient to have the prompt indicate the current directory for us. That way we always know "where we are" in the directory structure. To make this happen, type the following command at the DOS prompt you have now (which we will assume is C>).

prompt pg [enter]

Notice that the prompt now appears as C:\>. It is telling you that the current directory is the root directory of the C drive. You will see as we go along, that as we create directories and change from one to another, the prompt will tell you which directory you are in (i.e., the current directory) Now back to making and removing directories. Only three commands are required: MD, CD, and RD.

MD is the command to Make (create) a Directory. CD is the command to change the current directory. RD is the command to remove a directory (once it is empty of files).

To make a directory called, for example, REPORTS, the command would be:

MD REPORTS [enter]

CD is the command to Change from one Directory to another.

20

For example, to change from a directory called c:\EXPENSES to a directory called C:\INCOME, the command would be:

CD C:\INCOME [enter]

To Remove a Directory, the command is RD. For example, to remove a directory called REPORTS. the command would be:

RD REPORTS [enter]

Note, however, that all files must be deleted from a directory before the directory can be removed.

Detailed Steps for Creating DOS Directories

Assuming for the moment that the only directory we have is the root directory, we cannot yet change (CD) to another directory. So, we will begin by making some directories, using the steps shown in Figure 7-4.

Fig. 7-4. Steps for Creating DOS Directories

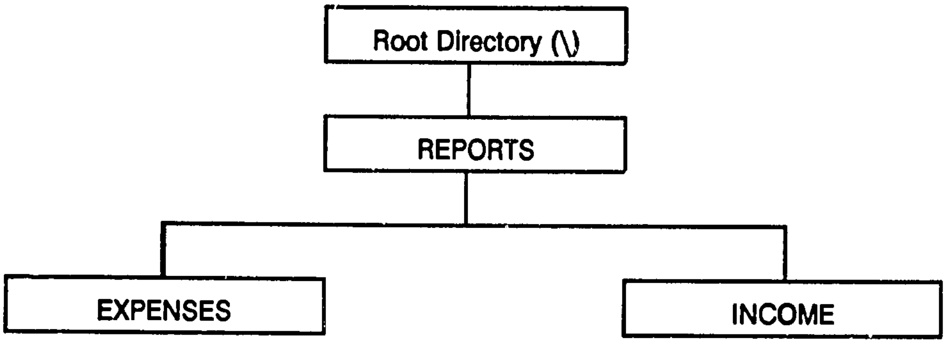
A. What we want too do:	B. The prompt now shows:	C. The command to be entered is:	D. The result is:
1. Make a directory called REPORTS, to be located immediately below the root directory	2. C:\	3. MD REPORTS (remember to press ENTER)	4. REPORTS directory has been created; prompt still shows C:\> because you are still in the root directory. When you make a new directory, it does not automatically become the current (active) directory!
5. Make a subdirectory called EXPENSES, below the REPORTS directory.	6. C:\	7. CD REPORTS Why? Because we must first change to the REPORTS directory, so we can make a directory right below it.	8. Prompt changes to C:\reports, which tells us that we are now in the REPORTS directory. REPORTS is now the current directory. Continue on the next row of this table.
9. Make a sub-irectory called EXPENSES, right below the REPORTS directory.	10. C:\reports	11. MD EXPENSES	12. Directory called EXPENSES now exists. Prompt does not change.
13. Make a subdirectory called INCOME, right below the REPORTS directory.	14. C:\reports This is where we need to be. We do not have to change to another directory.	15. MD INCOME	16. Directory called INCOME now exists. Prompt does not change.

In Figure 7-4, above, the column headings (A-D) indicate: (A) the desired task; (B) the current status of the DOS prompt; (C) the keystrokes to be entered to accomplish the task; and (D) the result of the keystrokes entered. To perform the exercise, follow the steps in order (1-16), as numbered in each cell of the figure.

212

Our directory structure is now as seen in Figure 7-5.

Fig. 7-5. Directory Structure from Steps in Figure 7-4



Practice in Changing the Current Directory

At this point we have a total of four directories: the root, REPORTS, EXPENSES, and INCOME. We can now practice changing from one directory to another, using the CD command.

If you performed the exercise given in Figure 7-4, your prompt right now should be C:\REPORTS>. Let's change to the root directory. The command is:

CD \

The prompt changes to C:\>.

Now let's change to the EXPENSES directory.

If you are guessing that the command is CD EXPENSES, it won't work. Why? Because DOS needs the "full" name of the directory, as we explained earlier. The EXPENSES directory is a subdirectory below the REPORTS directory. And, the REPORTS directory is a subdirectory of the root (\) directory. In a manner of speaking, you must "go through" the root directory and then the REPORTS directory to get to the EXPENSES directory.

So, the correct command is:

CD \REPORTS\EXPENSES

When you enter the above command, the prompt will change to:

C:\REPORTS\EXPENSES>

Indicating that you are now "in" the EXPENSES directory – the current directory is now EXPENSES.

Now let's change to the INCOME directory. The command is:

CD C:\REPORTS\INCOME

When you enter the above command, the prompt will change to:

C:\REPORTS\INCOME>

Indicating that you are now "in" the INCOME directory – the current directory is INCOME.

Shortcuts to Changing the Current Directory

As you can see, moving around between directories can require a bit of typing. There are two faster ways of doing it. Right now, you are sitting in the INCOME directory, a directory that is two levels below the root directory:

```
 ROOT
   |
REPORTS (one level down)
   |
INCOME (two levels down) <—— you are here
```

You can jump up to the level immediately above you by simply typing:

CD.. (CD followed by two periods or "dots")

Try it. The prompt will change to C:\REPORTS>.

Do it again. The prompt changes to C:\>. You have moved up two levels in all, and are now at the top of the directory structure.

Moving downward one level at a time is not quite as simple, but at least you don't have to type the "full" directory name. DOS will change you to a directory one level below where you are, if you just type CD plus the directory name. Let's try it.

CD REPORTS

Prompt changes to C:\REPORTS>.

Now we can go down one more level, either to the EXPENSES directory or the INCOME directory. Let's try EXPENSES.

CD EXPENSES

Prompt changes to C:\REPORTS\EXPENSES>. Notice that from here you cannot get to the INCOME directory by using either CD.., or CD\INCOME. Why?

The answer is that INCOME is on the same level with EXPENSES, where you are now. To get to it, you must type CD\REPORTS\INCOME.

Now let's get rid of these directories. We'll start with the INCOME directory. The first order of business is to remove any files stored in the INCOME directory. We'll assume that you have copied all of the files you wish to keep into another directory or onto a floppy disk.

First, change to the INCOME directory:

CD \REPORTS\INCOME

Prompt changes to C:\REPORTS\INCOME>.

Erase (delete) any and all files:

ERASE *.* (or, DEL *.*)

As explained in detail further on in this chapter, a DOS file name has two parts. The first part can have up to eight characters. The second part (suffix or extension) can have up to three characters. The two parts are separated by a period (also called "dot"). The asterisk (*) is called a "wild card" (because in the card game of "Poker," a "wild" card may be substituted for any other card in your hand).

When you type an asterisk (also called a "star") instead of the characters of a file name, you are telling DOS that you mean "any and all characters," regardless of what they are. Since the file name has two parts, you need an asterisk for both the first part and the second part, separated by a period.

Thus, *.* means "any and all files." This command is commonly read as "star dot star." In response to DEL *.*, DOS will query:

All files in directory will be deleted!
Are you sure (Y/N)?

If you are sure, type Y and press ENTER.

If files are then deleted, DOS does not tell you so. Check the contents of the directory by typing:

DIR (then press ENTER)

Removing Hidden Files and Read-Only Files

If the directory is empty of files, none will be listed. Ignore the "two files" message. These files are (.) and (..). They are not your files. If they are all that is shown for the directory, the directory is empty, except possibly for hidden files or "Read Only" files. Hidden files may be revealed by using the DOS ATTRIB command, as follows.

ATTRIB -H *.*

This will reveal all (*.*) of the hidden files in the current directory, and they can then be deleted. Similarly Read-only files can be changed to read-write files by the command:

ATTRIB -R *.*

If you enter the command DEL *.* and the directory is already empty, DOS will say: "File not found."

You are now ready to remove the INCOME directory. To do so you must move up one level in the directory structure:

CD..

The prompt changes to C:\REPORTS>.

Now remove the INCOME directory. The command is:

RD INCOME

Verifying the Removal of a Directory

How can we tell that the directory has actually been removed? There are two ways. First, type DIR (and press ENTER). If the INCOME directory still exists, it will appear as INCOME <DIR> in the REPORTS directory listing.

The other way is to attempt to change to the INCOME directory:

CD INCOME

If a directory called INCOME does not exist immediately below where you are (you are in the REPORTS directory), DOS will say: "Invalid directory."

Note that it is permitted to have more than one subdirectory of a given name within your directory structure. For example, you could have a directory called EXPENSES in two places, one as a subdirectory under REPORTS as we have done here, and another as a subdirectory somewhere else. But, you cannot have two subdirectories of the same name under the same "parent." If you have created subdirectory EXPENSES below REPORTS, you cannot create a second directory named EXPENSES under REPORTS. DOS will not allow it.

Further Discussion on File Names in DOS

Kinds of Files

There are essentially two kinds of "files," "program" files and "data" files.

1. A "program" file is a set of instructions the computer will execute. These are also called "executable" files, because the computer can execute (run) them, or "command" files because they consist of commands (instructions) that the computer is to execute.

   program file = executable file = command file

   An "application" is one or more program files, usually accompanied by a menu and designed for a specific purpose, e.g., word processing, spreadsheet, graphics. An application usually will automatically set up the function keys (keys F1-F10 or F1-F12, depending on your keyboard) to perform specific functions while you are running that application.

   program files + menu + function keys = application

2. A "data" file is a "product" (document, spreadsheet, database) you create by using the various programs (applications) such as WordPerfect, Lotus 1-2-3, etc. A data file, such as a letter you create with Word Perfect, or a spreadsheet you create with Lotus 1-2-3, consists of information that you enter and manipulate by using a program file. It is important to be able to differentiate and manage kinds of files according to how they are named, as explained below.

File Names

An important function of DOS is managing files. This includes naming files, copying them, deleting them and backing them up. A file name has a specific structure. On a computer running under DOS, a file name may contain up to eight characters, then a dot, then (up to) three characters after the dot. This applies to all files – program files, data files and other files.

The characters ahead of the dot represent the unique name of the file. The characters following the dot (called the file name "extension") are generally used to indicate the kind of file it is. (Note that not all file names have an extension, and in many cases an extension is not needed).

\- - - - - - - - . - - -

(file name, 8 chars) (dot) (extension, 3 chars)

Thus, program files (files the computer can execute) will have a name ahead of the dot, and then usually one of the following extensions:

filename.EXE	Example: 123.EXE (Lotus spreadsheet program file)
filename.COM	Example: COMMAND.COM (DOS file that interprets your keystrokes so that you can enter and use DOS commands)
filename.PRG	Example: MENU.PRG – perhaps a dBase program you wrote that displays a menu and allows the user to select options and other program files

Data files are a little different. Ordinarily, you give the file a name when you create it, but the application you are using will automatically tack on an extension when you save the file. This is simply a way of identifying the data files to help identify them by type, and by the application in which they were created. For example,

218

when you create and save a Lotus 1-2-3 spreadsheet as "Joe," the application will save it as joe.wk1. The ".wk1" indicates it is a Lotus 1-2-3 spreadsheet. When you create a database file in dBase, and name it "Joe," dBase will save it as joe.dbf, to indicate it is a database file, as opposed to any of the other kinds of files you can create or use in dBase.

Word Perfect does not automatically add an extension to a document when you create it and save it. You can use any extension you wish, perhaps to further identify a group of working documents. For example, you could tag all letters you write in September as filename.SEP, for October, filename.OCT, etc. You could then find, list, copy, delete and otherwise manage your files as a group such as "all September, all October, etc."

To do so, you would want the computer to ignore the file name (ahead of the dot) and fetch all the files ending in a particular extension, such as .SEP. This is done using "wild cards" as explained below.

Wild Cards

The term "wild card" comes from the game of poker and other card games in which any card specified as "wild" can be used in place of any other card. Thus, in poker, if "deuces" are declared to be wild cards, then a hand having two aces and three deuces could be declared to "really" be five aces, because you can make the deuces be any card you wish. Since five aces is the best hand you could make, you would call the deuces "aces" on that occasion. This analogy to DOS, however, is somewhat inaccurate. If you do not play poker, forget about aces and deuces, and read on.

The Wild Card (*) for Groups of Characters

In DOS, the asterisk (*) is used when you want the computer to IGNORE the characters that come either before or after the dot, and to focus on the remaining characters.
By "ignore" we mean "include any and all, without differentiating on the basis of the group of characters the asterisk has replaced."

For example, you may have 100 word processing documents, of which 20 have the extension .SEP. You want the directory to list the .SEP documents on the screen. It would be laborious, for example, to hunt for:

Mary.SEP
Joe.SEP
Fred.SEP
Mike.SEP

219

Instead, you would indicate "any and all" filenames by using an asterisk before the dot, and the .SEP extension after the dot. Thus the command:

DIR *.SEP

will produce a list of all files ending in .SEP, regardless of the names they have ahead of the dot.

Suppose you wanted to copy all files from a floppy disk into a directory on your C drive. In essence, you are saying you want to copy a file regardless of its filename head of the dot and regardless of its extension characters that follow the dot. To do this, you would just replace the filename portion with an asterisk and you would replace the extension portion with another asterisk. So, the basic command would be:

COPY *.*

To have a complete command, you need to specify the location of the files to be copied and the location to which they are to be copied.

Suppose you want all files on a floppy disk in the A: drive to be copied into a directory C:\WP51\STUFF. The command would be:

COPY A:*.* C:\WP51\STUFF [RETURN]

Instead of saying "asterisk," we simply say "star." So, this command is read "copy star-dot-star."

Remember: a command word such as COPY, DIR and ERASE is always followed by a space. But, never put a space after the colon in a drive name, and never use spaces as part of a file name (including when you use an asterisk) or a directory name (path).

Suppose you wanted to list all files beginning with the letter "S" regardless of the extensions they may have. The command would be:

DIR S*.*

Notice there is a space after the word DIR but no spaces in the string S*.* The computer will search for all file names beginning with "s," and it will then include any and all files regardless of the letters that fall between the "s" and the dot. It will include any and all files, regardless of the letters comprising their extensions.

The Individual Character Wild Card (?)

As you may have noticed, the star is used in place of groups of letters rather than to replace an individual letter in a file name or extension. There are times, however, when you will need a wild card for individual letters. The DOS wild card for an individual character is the question mark (?), as explained below.

Suppose you have 500 files on your computer and you are looking for those files named either SMITH OR SMYTH, and you don't care what the extensions are. The third character is the culprit. We need to replace it with a "?" so it will be ignored in the search, as follows:

DIR SM?TH.*

This command says, show me all files beginning with SM and ending with TH, regardless of what the third character is, and regardless of the file name extension.

The question mark is also useful where you want to identify files having only a certain number of characters in their file name. Perhaps you are looking for a file that has only two letters in its name, but you can't remember what they are.

The command would be:

DIR ??.*

This command will list all files having a two-letter name, regardless of the extension.

Self-Test on DOS Wild Cards

Can you figure out what the following commands would do? (Answers are at the end of this chapter.)

1. DIR *.COM
2. DIR ?EST.*
3. DIR RES?.BAT
4. ERASE *.DOC
5. COPY *.* A:
6. COPY A:*.* C:\LOTUS\DATA

Listing Directories

Like files, directories have names. But the name of a directory does not have an extension. So, if you wish to see a list of your directories rather than the files within the directories, the command would be:

DIR *.

Notice that there is nothing after the dot, because directories have no extension and therefore nothing is needed. Notice also that the star says, "any and all names," but you have specified nothing after the dot. DOS recognizes this as a request to list only the names of directories (and any files that have no extensions). The directories, however, will be distinguished by "<DIR>," which follows their name on the display.

If, instead, you enter, DIR *.*, all immediately subordinate directories and all files in the current directory will be listed.

Answers to Self-Test on DOS Wild Cards:

1. List files in the current directory that end with .COM.

2. List files in the current directory that have four-letter names, of which letters 2-4 are e,s,t.

3. List files in the current directory that have four-letter names, of which letters 1-3 are r,e,s, and the extension is BAT.

4. Erase all files in the current directory that end with .DOC.

5. Copy all files in the current directory onto the (A:) drive.

6. Copy all files from the (A:) drive into the subdirectory named "data," which is a subdirectory under the directory named "Lotus," which is a subdirectory of the root directory (\) on drive (C:).

222

NOTES

8

DECIMAL, BINARY AND HEX NUMBERS

CHAPTER 8: DECIMAL, BINARY AND HEX NUMBERS

The Basis of Numbering Systems

To understand how decimal, binary and hex (hexadecimal) numbers work, it will be helpful first to consider briefly a different kind of number system, the ancient system of Roman numerals. In contrast to other systems of numbers, Roman numerals do not use the position of a digit (such as a column for units, tens, hundreds, and so on) to indicate its value.

Just by looking at the first ten Roman numerals, it is easy to see that the position of a given numeral from left to right, or right to left, has no significance. A value from one to ten, for example, is represented by anywhere from one to three characters (I,V,X), occupying anywhere for one to four columns (positions).

I
II
III
IV
V
VI
VII
VIII
IX
X

Since the position of a character is not used to help define the overall value of the number written, it is all but impossible to do arithmetic using Roman numerals. Imagine, for example, having to multiply XXIV by XXXVIII, or divide MCX by XIX.

Other numbering systems are designed according to a different principle. They do use the position of the digits to indicate the overall value of the number written. It is important to understand this idea of "position coding," because it will make hexadecimal and binary numbers much easier to understand when we deal with them in just a moment.

To convert from binary and hexadecimal numbers to their decimal number equivalents, we will be using a tabular format. The basic idea behind the conversion charts is quite simple. The first row of the chart indicates the value of the column; the second row indicates "how many" events occur for each column. To illustrate, consider Figure 8-1, which might have been created to report how much cash was on hand in a store. The first row indicates the denomination of the bills on hand. The second row indicates the quantity of each denomination on hand.

Fig. 8-1. Frequencies and Denominations of Currency

Denomination of Bill:	$100	$50	$20	$10	$5	$1
Number of Bills:	32	16	126	75	25	37

We would have no difficulty in seeing that the total amount of cash on hand is found by multiplying the number of bills in each category by its denomination, and then adding up all categories to obtain the total, $5164.

32	x	$100	=	$3200
16	x	$50	=	800
126	x	$20	=	252
75	x	$10	=	750
25	x	$5	=	125
37	x	$1	=	37
				$5164

This same basic notion applies to systems of numbers where each digit or character takes its value from the column (position) it is in, and where the value of the columns is determined by the number of characters in the counting system. Thus, in our normal system of counting (decimal system) we recognize the character "1" as being a "one" when it stands alone, but a "ten" when shown in the second position (10), one hundred when shown in the third position (100), and so on. This concept is carried forward below.

The Decimal System

In our normal system of counting, the decimal system, we have ten characters to use: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. We begin counting, and when all of the characters are used up, we increment the position immediately to the left by one, and begin again.

226

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
...
18
19
20
21
22
23
...
27
28
29
30
31
32
...
37
38
39

...and so on, until we have used all of the characters in the second column as well:

90
91
92
...
97
98
99

At this point we begin a third column to the left, and start again with the first two columns set to zero:

100
101
102
103
...
197
198
199

200
201
202
203
...
297
298
299

300
301
302
303
...
397
398
399
...
997
998
999

When the third column has use all of the characters, we begin a fourth column to the left and reset the first three columns (positions) to zero:

1000
1001
1002
1003

Because we are so accustomed to using this system, we no longer stop to think about the importance of the position of each character. Thus, we read a number such as 1354 immediately as "one thousand, three hundred and fifty-four."

As Figure 8-2 illustrates, we are actually reading the value of each character according to the position it occupies in the sequence.

Fig. 8-2. Position Values in the Decimal System of Numbers

Thousands (1000s)	Hundreds (100s)	Tens (10s)	Units (1s)
1	3	5	4

Figure 8-2 indicates we have a total quantity consisting of:

1	thousand(s)	=	1000
3	hundreds(s)	=	300
5	ten(s)	=	50
4	one(s)	=	4
	Total	=	1354

Note that the digits 1 3 5 4 merely indicate the frequency or quantity of items occurring in a given column (position). And as we shall see, the value of each column or position is determined by the number of characters we have to work with.

It is essential to understand that the characters we use (0,1,2,3,4,5,6,7,8,9) are purely arbitrary, and it is only by historical accident that we use ten such characters rather than, say, five or eight or sixteen. It has been suggested that this came about because we have ten fingers and ten toes, and so the decimal (meaning a base of 10 characters) system came into being quite naturally.

Using the decimal system, for example, we count the following objects and report that there are 114 of them. But the digits 114 represent the total number of objects only because we all agree they have been counted using the decimal system.

But what if we had only five fingers and five toes? What if the characters 0,1,2,3,4, had been invented, but 5, 6, 7, 8, and 9 had not? We could still have a valid counting system, and it would work the same way. We would follow the same procedure as described above, using up the characters in each column and beginning a new column when we run out of characters, as follows:

0
1
2
3
4

Out of characters, so start a column to the left, and reset the first column to zero:

10
11
12
13
14

Out of characters again, so increment the left-hand column by one and begin again:

20
21
22
23
24

Out of characters again, so increment the left-hand column by one and begin again:

30
31
32
33
34

Out of characters again, so increment the left-hand column by one and begin again:

40
41
42
43
44

Now out of characters for the first and second columns, so start a third column and reset the first to columns to zero:

100
101
102
103
104

Now out of characters for the first column. So, increment the second column by one, and start again in the first column:

110
111
112
113
114

and so on...

In our familiar, decimal system, the digits 114 would represent one "hundred," four "tens" and four "ones," for a total of one hundred and fourteen objects (days, dollars, or whatever). But here, we have only five characters. In this case, the digits 114 must be read from a different frame of reference, because we have not ten but only five characters to work with, as shown in Figure 8-3.

Fig. 8-3. Position Values in the Base 5 System of Numbers

Twenty-fives (25s)	Fives (5s)	Units (1s)
1	1	4

Notice that the column (position) values are not units, 10s, 100s and so on as they are in the decimal system. Rather they are units, 5s and 25s, because we have only five characters to work with. The number of characters is what determines the value of the column (position). Thus, the digits 114 here (in our system limited to five characters) represent only the following number of objects.

So, the digits 114, for example, have no absolute meaning unto themselves. The quantity of physical objects these digits represent depends entirely on the system of counting from which they came. To make sense of this, we can relate the digits in any system back to our more familiar decimal system. Thus, the digits 114 derived in the base five system, just shown, represent a quantity of objects that in the decimal system would be represented by the expression "34."

In more concise form, we would state: 114 base 5 = 34 decimal.

Figure 8-4 shows decimal (base 10) numbers 0 through 34 and their base 5 equivalents.

In summary, there are only two main things to be concerned about in understanding any counting system. First is the number of characters the system has. Second is that the number of characters determines the value of the columns (positions) the characters occupy. In the preceding example, we have seen what happens when we reduce the number of characters from ten to five. Now let's examine a system that uses only two characters— the binary (or, "base 2") system of numbers.

Fig. 8-4. Decimal Numbers 0-34, and their Base 5 Equivalents

Decimal	Base 5	Decimal	Base 5
0	0	18	33
1	1	19	34
2	2	20	40
3	3	21	41
4	4	22	42
5	10	23	43
6	11	24	44
7	12	25	100
8	13	26	101
9	14	27	102
10	20	28	103
11	21	29	104
12	22	30	110
13	23	31	111
14	24	32	112
15	30	33	113
16	31	34	114
17	32		

The Binary System

"Binary" means "two." In the binary system, we have only the characters 0 and 1 to work with. Otherwise, the procedure for counting is exactly the same as we have just seen for decimal numbers and base 5 numbers. As we go along below, we show the decimal equivalent at the right of the binary number.

Binary	Decimal
0	0
1	1

Out of characters, so increment column at the left and set first column to zero:

Binary	Decimal
10	2
11	3

Out of characters for both columns, so start a third column and reset the first two columns to zero:

Binary	Decimal
100	4
101	5
110	6
111	7

Out again, so begin a fourth column at the left, and set the first three columns to zero:

Binary	Decimal
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Out again, so begin a fifth column at the left, and set the first four columns to zero:

Binary	Decimal
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26

and so on...

As we said earlier, the physical quantity of objects represented by a string of digits or characters depends entirely on the counting system from which those digits or characters were derived. Here for example in the binary system it should thus be no surprise that the digits "100" (one, zero, zero) do not represent "one hundred" objects. The digits 100 take their meaning here from the column (position) values of the binary system, as shown in Figure 8-5.

Fig. 8-5. Positional Values In the Base 2 (Binary) System of Numbers

Sixteens (16s)	Eights (8s)	Fours (4s)	Twos (2s)	Units (1s)

If we put our binary number, 100, into the chart (Figure 8-6), we see that it represents one 4, no 2s and no 1s, for a total of 4 objects.

Fig. 8-6. Using a Chart to Convert Binary Numbers to Decimal Numbers

Sixteens (16s)	Eights (8s)	Fours (4s)	Twos (2s)	Units (1s)
(none)	(none)	1	0	0

Therefore, 100 binary = 4 decimal. But, be careful. The foregoing statement must not be read as, "one hundred binary equals four decimal." It should be read as, "one, zero, zero binary equals four decimal." A common error is to refer to numbers from one system by the name given the same character string in a different system. This practice should be avoided, as it causes mistakes when converting numbers from one system to another.

Binary numbers are often written with leading zeros. The leading zeros may be ignored, in that they have no effect on the number they precede. For example, the binary numbers 100, 0100, 00100, and 000100 all are the same number, and each is equivalent to 4 in the decimal system.

What is the decimal equivalent of binary 01101? Figure 8-7 illustrates how to obtain the answer. Each binary digit is simply multitplied by the decimal value of the column it occupies.

234

Fig. 8-7. Converting a Binary Number to Its Decimal Equivalent

Sixteens (16s)	Eights (8s)	Fours (4s)	Twos (2s)	Units (1s)
0	1	1	0	1

How many 16s? None. 16 x 0 = 0
How many 8s? One. 8 x 1 = 8
How many 4s? One. 4 x 1 = 4
How many 2s? None. 2 x 0 = 0
How many 1s? One. 1 x 1 = 1
Total (decimal) = 0 + 8 + 4 + 0 + 1 = 13

Therefore, 01101 binary = 13 decimal.

Thus far we have seen three different systems – decimal, base 5 and binary (base 2). In each case, the basic rules were the same; only the number of characters allowed was different. We also saw that any given string of digits or characters has meaning only when the system from which it came is specified. Just as we have seen two systems that use fewer than ten characters, so are there systems that use more than ten characters. One such system of importance to the field of computers is the "hexadecimal" system, as described below.

The Hexadecimal System

The term "hex" (or "hexa") means six, and decimal means ten. So, the hexadecimal system ("hex" for short) contains six plus ten, or 16 total characters. They are the usual ten decimal characters (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) and the letters A, B, C, D, E, and F. A word of caution – just because the ten decimal characters are included in the hexadecimal system, it does not mean that a string of such digits can be read the same as if they were occurring in the decimal system of counting.

Remember that the characters (digits, letters) take their value from their position, and that the value of the position depends on the number of characters used in the system. Thus, the digits 100, for example, in hexadecimal do not represent the same quantity of objects as would be true in the decimal system.

The rules for counting and incrementing columns in the hexadecimal system are the same as we have seen for the other systems discussed above: Use all of the characters, then increment the column at the left by one, and begin again. In this

235

case, we continue past the digit 9, and use six more characters, A, B, C, D, E, F. Figure 8-8 shows the hex numbers and their decimal equivalents from zero through 34. As shown, hex A is equivalent to decimal 10; B is equivalent to 11, C is equivalent to 12; D is equivalent to 13; E is equivalent to 14; F is equivalent to 15. At that point, the hex sequence of characters starts over again.

What is the decimal equivalent of hexadecimal F6? Once again, we can use a chart to easily solve the problem of conversion. Recall that the value of the columns (positions) is determined by the number of characters in the system.

In the decimal system, the positions increase by a factor of 10; in the base 5 system, they increase by a factor of 5; in the binary system, they increase by a factor of 2. Since we have 16 characters in the hexadecimal system the value of each position increases by a factor of 16, as shown in Figure 8-9.

We have already seen that the letter F in hexadecimal is equivalent to 15 decimal. Note in the chart that the F appears in the 16s position. Therefore, we have fifteen "16s" as part of the number we are concerned with. We see, also, a 6 in the 1s position. The decimal number we are after is simply the sum of these values:

F x 16 = 15 x 16 = 240

6 x 1 = 6

Total = 240 + 6 = 246 decimal

Therefore, F6 hexadecimal = 246 decimal.

Let's try a more difficult example. What is the decimal equivalent of the hex number D3A4?

Let's put the number into a chart, as shown in Figure 8-10, and see what it adds up to. See if you can solve it before turning to page 8-14 to find the answer.

To help you along, the following equivalents apply:

D hexadecimal = 13 decimal
3 hexadecimal = 3 decimal
A hexadecimal = 10 decimal
4 hexadecimal = 4 decimal

23.10

Fig. 8-8. Hexadecimal Numbers and their Decimal Equivalents

Hexadecimal	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15
10	16
11	17
12	18
13	19
14	20
15	21
16	22
17	23
18	24
19	25
1A	26
1B	27
1C	28
1D	29
1E	30
1F	31
20	32
21	33
22	34

Fig. 8-9. Converting a Hexadecimal Number to Decimal Number

65,536s	4096s	256s	16s	1s
(none)	(none)	(none)	F	6

Fig. 8-10. Converting Hexadecimal D3A4 to Decimal

65,536s	4096s	256s	16s	1s
(none)	D	3	A	4

The solution to the problem posed on page 8-12 is solved using Figure 8-10, as follows.

In the 4096s column we have a D, and D equates to 13 decimal.
Therefore, we have 13 x 4096 = 53248 decimal.

In the 256s column, we have a 3.
Therefore, 3 x 256 = 768 decimal.

In the 16s column we have an A. "A" in hex is 10 in decimal.
So, 10 x 16 = 160 decimal.

In the 1s column we have a 4.
4 x 1 = 4 decimal.

Add them all up:

53248 + 768 + 160 + 4 = 54,180 decimal

Therefore, D3A4 hex = 54,180 decimal.

Hexadecimal Number Indicators

Since one may encounter numbers from different systems, it is customary to indicate the type of number being presented. Hexadecimal numbers are indicated by "H" or "h," which may or may not be accompanied by an ampersand (&), in any of the following ways:

D3A4H
D3A4h
&HD3A4
&hD3A4

All are the same hexadecimal number, D3A4. Neither the letter "H" (or "h") nor the ampersand (&) have any effect on the value of the number indicated. Also, as with binary numbers, hexadecimal numbers are often written with leading zeros. The leading zeros do not affect the value of the number presented. Thus,

00D3H
00D3h
&H00D3
&h00D3

are all the same hexadecimal number, D3.

Chapter Summary

In this chapter we have examined the principles of number systems that use the position of their characters (digits, letters) in specifying the numeric quantity represented. In these systems, we have seen that the value of the position is determined by the number of characters used in the system, and that the value of the columns (position) increases from right to left by a factor (multiple) equal to the number of characters used in the system.

We have demonstrated by use of conversion charts how a number in one system may be translated into its equivalent expression in another system. Further, we have cautioned against referring to an expression in a given system by a name applied to it in another system. For example, a binary number such as 100 should not be referred to as "one hundred" binary, since the term "hundred" implies use of the decimal system and it has no meaning in the binary system.

Similarly, one should not refer to a hexadecimal number such as AF55 as, "A F fifty-five." Again, the term "fifty-five" is an expression applying to decimal numbers. In systems other than decimal, the numbers should be read as individual characters (or digits) to avoid errors and confusion when working with or converting such numbers. Finally, remember that hexadecimal numbers are indicated using "h" or "H," which may or may not be accompanied by an ampersand (&); and that, in all cases, leading zeros may be ignored insofar as they have no affect on the value of the number presented.

As a review, Figure 8-11 presents the decimal numbers 0 through 32, and their equivalent expressions in binary, hexadecimal, and base 5. By examining the figure, it should now be easy to see how each system cycles through its characters, and propagates positions to the left as the numbers increase in value.

Fig. 8-11. Some Decimal, Binary, Hexadecimal and Base 5 Equivalents

Decimal	Binary	Hex	Base 5
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	10
6	110	6	11
7	111	7	12
8	1000	8	13
9	1001	9	14
10	1010	A	20
11	1011	B	21
12	1100	C	22
13	1101	D	23
14	1110	E	24
15	1111	F	30
16	10000	10	31
17	10001	11	32
18	10010	12	33
19	10011	13	34
20	10100	14	40
21	10101	15	41
22	10110	16	42
23	10111	17	43
24	11000	18	44
25	11001	19	100
26	11010	1A	101
27	11011	1B	102
28	11100	1C	103
29	11101	1D	104
30	11110	1E	110
31	11111	1F	111
32	100000	20	112


9
PREVENTIVE MAINTENANCE

CHAPTER 9: PREVENTIVE MAINTENANCE

What Is Preventive Maintenance?

For our present discussion we shall consider preventive maintenance (PM) to be any action taken to reduce the likelihood of a malfunction or failure of a system or its components, and/or to prolong the operating life of a system and its components. The primary purpose of preventive maintenance is to prevent events or conditions that would cause damage from which the system components and/or data could not be restored.

Paramount among these events is the permanent loss of data. For this reason, we will devote considerable attention in this chapter to doing backups. Maintenance should not be performed on system unless the system has been backed up. Since a malfunction may prevent a backup from being done, all systems should be backed up routinely, according to a prescribed schedule. <u>This is the most important preventive maintenance that can be performed</u>.

Next is the loss or unavailability of data, components or functions critical to immediate operations. Next is the malfunction of items that are difficult and/or expensive to repair or replace. The hierarchy continues downward, through lesser degrees of inconvenience and expense due to failure or malfunction of system components.

Despite its importance, preventive maintenance is often not given the attention that it deserves. One reason for the neglect of preventive maintenance is that personal computers are in general quite reliable, and so they do not seem to need preventive maintenance. Further, users do not like to be deprived of their machines while PM is being performed. Without a positive attitude toward PM and a well structured plan to support it, PM will be at best a casual and relatively ineffective activity. The penalty will invariably be one or more of the undesirable events listed in the preceding paragraphs.

An effective maintenance program is formal, scheduled and documented (recorded). Scheduling may be based on elapsed time or on cumulative hours of system use. Of course, preventive maintenance may also be performed at any time symptoms arise, such as when a disk drive begins to squeak. Further, preventive maintenance is based on prescribed policies and procedures that ensure the program will be carried out and that actions taken will be timely, correct and complete. Therefore, preventive maintenance requires support from senior management and cooperation from the users it serves.

The alternative to an effective maintenance program is to gamble (or to pray or hope) that nothing wrong will ever happen. Since malfunctions are unavoidable, such a gamble is ill advised. The only question is not <u>whether</u> something will go wrong, but <u>when</u> it will go wrong.

242

Items to be Maintained

Those familiar with "Murphy's Law" know that if something does go wrong, it will do so at the worst possible time. A corollary might well be that a poor program of preventive maintenance is worse than no program at all, because a poor program runs the risk of providing a false sense of security. Unfortunately, it often takes a disaster to convince people that preventive maintenance pays a high return on the investment made in it.

Having said all of that, the matter at hand is after all not very difficult, and most of it is based on common sense. What are the things to be maintained? Here is the primary list, and it is not very long:

- Environment
- Keyboard
- Adapters
- Monitor
- Cables
- Motherboard
- Disk drives
- Printer
- Power Supply

Maintenance Steps

There are five main steps for the performing effective preventive maintenance on a personal computer:

1. Inspection
2. Documenting
3. Cleaning
4. Lubricating
5. Adjusting

The first step is critical, because without inspection, impending problems cannot be discovered. The second step, "documenting," is vital because without it one cannot know what has and has not been done, and a schedule cannot be maintained. Further, without a record, one cannot learn effectively from past performance and problems. Steps 1 and 2 apply to all aspects of the system and its environment.

43

Which of the remaining steps then apply to which parts of the system? Opinions may vary, but Figure 9-1 provides a basic profile. The five steps shown across the chart are as defined above (inspection, back up, cleaning, etc.). An "X" in a cell indicates that the maintenance step in that column applies to the item listed in that row. Note that many the cells do not contain an X. Most items require only cleaning and minor adjustments. Units such as power supplies, monitors, adapters and printers are not taken apart for preventive maintenance. Indeed, other than cleaning, they require little or no preventive maintenance.

Fig. 9-1 Summary Chart of Personal Computer Preventive Maintenance

	Maintenance Steps*				
Item Maintained	1	2	3	4	5
Environment	X	X		X	X
Keyboard	X	X			X
Monitor	X	X			X
Cables	X			X	X
Disk drives	X	X	X	X	X
Adapters	X	X			X
Motherboard	X	X		X	X
Power Supply	X	X			X
Printer	X	X	X	X	X

*1 = Inspection; 2 = Cleaning; 3 = Lubricating;
4 = Adjusting; 5 = Documenting

Since inspection and documentation apply to all aspects of PM, we will address them first, as general topics. For conducting an inspection, a series of illustrative questions are posed. These and others, as appropriate to the specific environment and equipment, may readily be converted into a handy checklist.

Documentation involves several steps, including preparation of an equipment inventory; obtaining system baseline configuration and performance records; preparing and maintaining a trouble log to be kept at each machine; preparing and maintaining a trouble call (service) log; and finally, creating and maintaining a preventive maintenance schedule and service/problem data base.

Inspection

Inspection involves looking (visual), listening (auditory), touching/feeling (tactile), and asking (verbal) activities. The first three apply to the equipment; the last applies to the system user or operator.

Visual Inspection

- Look at the working environment. Is it neat and clean, or is it cluttered and disorganized?

- Are papers stacked on top of the monitor or around the computer case, thus interfering with ventilation and cooling? Are cables stretched or cramped because of their position on the desk or workstand? Are numerous devices plugged into a single extension cord (or multi-outlet surge protector), or are too many devices sharing the same electrical line (e.g., several computers, printers, FAX machine, coffee maker?), thus inviting electrical power problems?

- Are there sources of electromagnetic or radio interference too close to the system? Are floppy disks strewn about or are they carefully labeled and stored in a proper container? Are there magnetic objects such as paper clip holders, magnetized scissors/tools, or wall magnets located where they can come into contact with floppy disks?

- Are floppy disks stacked on or close to the display monitor, where they will be exposed to a strong electromagnetic field when the display is powered on?

- Is the system or its magnetic media exposed to direct sunlight or other source of heat?

- Is the area clean, or is there evidence of smoking, food and/or beverages, any of which might contaminate the system or its magnetic media?

- Examine the general condition of the equipment. Is it clean or is it covered with dirt or dust? Are any surfaces worn or chipped? Are any cables crimped, cut, frayed or otherwise in poor condition?

- Are all of the required cover screws, adapter board retaining screws, and connector plug retaining screws present?

- Are all of the luminous displays and indicators functioning properly (keyboard, front panel, disk drives, printer panel)?

A more detailed visual inspection would include examining the status of the hard disk drive (is it approaching full capacity?). The DOS CHKDSK command will provide this information. The DOS MEM command (in DOS version 5.0) will display the details on system memory; note whether the system has been loaded up with TSR (Terminate and Stay Resident) programs that may invite memory management problems.

Auditory Inspection

- If something is supposed to make a sound, does it do so, and is the sound normal? Examples are system speaker; speaker on communications board; disk drive whir and chatter; and keyboard "clicks."

- Are there any unusual sounds? Are there any squeaks, whines, or vibration noises (especially disk drives)?

Tactile Inspection

- Sense (or measure) the temperature and humidity. Is it within the specified range for the equipment? The ambient (surrounding) temperature should be between 60 and 80 degrees Fahrenheit (16 - 27 degrees Celsius).

- An inexpensive digital probe can be used to check the operating temperature inside the computer. The inside temperature should not exceed 110 degrees Fahrenheit (43 degrees Celsius).

- Is there evidence of electrostatic discharge because of low humidity and presence of carpeting or other materials that contribute to static buildup?

- Shuffle your feet across the floor, and then touch a metal doorknob. The answer should be evident.

- Is anything loose that is supposed to be tight? Touch and feel the items. Are connections secure and fastened with the screws or clips provided?

Verbal Inspection

- Ask the user whether there have been any problems or whether anything unusual has been occurring with the system. Has anything been added, removed or modified (memory chips, boards, peripherals, software?) Has anyone serviced the machine since it last had PM? Has the machine been moved, for example, carted from one place to another – was the machine prepared correctly for moving?

Documenting

A major challenge to any organization is in keeping track of the computer equipment it has. Knowing exactly what equipment exists; where it is physically located; what the configuration (i.e., installed options), age, condition, and warranty (or service contract) status is; and what the current value of the equipment is (e.g., for asset and insurance purposes) collectively represent a fairly large administrative task. It is not unusual, therefore, when the organization is faced with writing a disaster recovery plan, performing a risk assessment and management analysis, or installing a comprehensive network, that the lack of an adequate inventory poses an immediate obstacle. Further, an inventory and configuration data base is needed in order to formulate and execute a competent program of preventive maintenance.

Inventory

The details of the overall task as described above are, from an organizational standpoint, beyond the scope of this book. However, the advent of diagnostic software packages (such as Checkit, as described in Chapter 11)) have greatly simplified the task of collecting much of the needed information. The procedure is simply to run the diagnostic (e.g., Checkit) on each machine. Run all of the features provided on the package (configuration information; tests; benchmarks), and save the results both to disk and as a printout. The package allows a line of descriptive information to be entered by the user, which can serve to uniquely identify the machine in question, as well as its present location. For example, the line might be entered as:

Serial No. 1234-003 Bldg. 6C, Rm. 409B User: J. Smith

The date and time that the test is run is automatically included in the report. Of course, the report will not include a description of the printer or other peripherals attached to the machine, but it does provide an excellent starting point for an inventory and maintenance management data base.

Baseline Records

The information obtained as described in the preceding paragraph also provides the baseline data for each machine. When a given machine is scheduled for routine maintenance or when it has a malfunction or complaint, the technical support person will have a record of how the machine is configured and how well it was performing before the problem arose.

Trouble Log (At the Machine)

The user of the PC should maintain a record of problems, unusual behavior by the machine, or anything done for or to the machine. The record should be kept at the machine, as a reference for the technical support person. Why is this important?

Problems do not always occur all of a sudden. Often, they appear gradually, but not of sufficient magnitude for a service call. The course of development of a problem often will provide a clue as to the source of the problem when the machine finally does fail. So, a record of any "suspicious behavior" on the part of the machine, or anything added to, removed from, or changed on the machine can save time, inconvenience and expense when troubleshooting is needed.

Preventive Maintenance and Trouble Call Log

A log (record) of all preventive maintenance and trouble calls should be maintained for each machine. Preventive maintenance and trouble call records may be kept on different forms, but they should not be isolated from each other. Both contain the same basic information regarding what was done, when and where it was done, and by whom it was done.

The only difference between the two (other than resolving vs. preventing problems) is that preventive maintenance is scheduled ahead of time, whereas trouble calls are serviced on demand. Obviously the effectiveness of a PM program should be evaluated in part by its ability to reduce the number of trouble calls experienced.

Given below are some of the key entries for the PM and trouble call log. The corresponding item for PM is shown in parentheses.

- Name and telephone number of person reporting the trouble (or scheduled for PM)
- Date and time trouble call is received (time and date of scheduled PM)
- Machine ID, and location
- Description of the complaint, symptoms or malfunction (checklist of PM to be performed)
- Date and time of service response: the attempt to resolve the problem (date and time PM was performed)
- Description of problem diagnosed (completed PM checklist, with problems or deficiencies noted)
- Corrective action(s) taken (same for PM, if PM uncovered any problems)
- Whether problem was satisfactorily corrected (whether PM was satisfactorily completed)
- Action taken/recommended to prevent recurrence of the problem (action taken/recommended to improve environment or other PM issues)
- Follow-up action planned, if problem could not be resolved (date and time of next PM performance)
- Name and telephone number of technical support person responding to the trouble call (name and telephone number of PM technician)

Maintenance Data Base

It is apparent from our discussion thus far, that a considerable amount of data can and should be collected in support of computer resource management, preventive maintenance and troubleshooting. All of that data, however, is of little use if it cannot be easily accessed, updated and queried. The answer is to put the data into a formal data base. It is assumed that most PC users today are familiar with at least one of the more popular data base applications (e.g., dBase; RBase; FoxBase), so we will not address here the many capabilities that these packages offer. Any one of those mentioned, as well as a host of others, would be suitable for creating a maintenance data base.

Since some of the data can be collected automatically by Checkit or a similar package, there is no need to copy (re-key or import) that data into the data base. It would probably be sufficient just to include a field in the data base that identifies the "Checkit" disk file or printout record for each machine. This is appropriate because it is unlikely that one would need to search or sort the data base by the variables contained in a configuration, test or benchmark report. Therefore, such information may as well be maintained in its original, document format.

For those less familiar with data base applications, it may be worth noting that word processing applications and spreadsheet applications are not adequate substitutes for a data base package. Many an inventory has been put into word processing (i.e., a document, or "flat file") or a spreadsheet, only to find that these applications cannot provide the searching, sorting (indexing) and reporting functions needed to use the "data base" efficiently.

With the forgoing overview of organizational issues, data collection, inspection and documentation as background, we shall now turn our attention to the objects of the PM as listed in Figure 9-1, and the details of preventive maintenance – cleaning; lubricating; and adjusting. As we shall see, not all steps apply to every item, so the overall job is not as large as it may first appear.

Cleaning, Lubricating and Adjusting

Here is a list of basic items needed for doing preventive maintenance on a personal computer. The exact items that the reader may need will depend on the machines to be serviced and the extent of the PM to be performed.

- Electronics cleaning solution (freon/alcohol)
- Distilled water
- Isopropyl alcohol (common "rubbing alcohol")
- Clean, lint-free cloth
- Soft brush (1/2 inch length bristles)
- Foam-tipped swabs (do not use "Q-Tip" type cotton swabs; the cotton fibers come off and fall into the machine or adhere to the components)
- Silicone lubricant

-250

- Conductive lubricant
- Computer (or scale-model) vacuum cleaner
- Can of electronics quality, compressed air. Do not use general purpose compressed air — it contains too much moisture, it may contain contaminants, and the nozzle velocity may be too high.
- Small, thin-blade screwdriver (for prying up keyboard caps)
- Screwdrivers for tightening retaining screws (on adapters and cable connector plugs)
- Electrostatic grounding strap
- Small tube of silicone rubber "bathtub" caulking (to dampen a noisy grounding strap on a hard drive)

Cleaning

Cleaning applies to the work environment in which the PC is used, and to the PC itself. By keeping the area (and the air) surrounding the PC clean, the need for cleaning the interior components will be reduced.

Cleaning the Work Environment. The environment should be kept free of dust, dirt and smoke. If these contaminants cannot be prevented from entering the work area, they should be removed regularly by manual cleaning, as is normally done in an office. Note, however, that vacuum cleaners, rug shampooing machines and similar devices are strong sources of electromagnetic and radio interference energy. They should not be used in proximity to a computer or magnetic storage device.

Cleaning PC Interior Components. The PC interior components are all of the things that reside inside the cover of the PC. Mainly the task of cleaning is no more than removing dust and dirt that has collected on the surfaces of the components.

Where does all of the dust come from? The system is cooled by outside air that is pulled through the system by a fan inside the unit. The air enters through vents in the computer cover and/or panels, and through the opening at the front of the floppy drive. The constant flow of air deposits dust and other airborne particles on the surface of components inside the machine. Dust is a good insulator. If a layer

251

of dust is allowed to accumulate inside the machine, it will prevent components from cooling properly, and so will shorten their operating life.

To clean the inside of the unit, the cover of course must first be removed. The preferred method of cleaning is to use computer quality, canned air (compressed air-in-a-can). The components are simply sprayed with air from the can, to displace and hopefully remove the dust from inside the machine. The trick is to keep the dust from just moving from one area inside the machine to another, especially to the disk drives. The best approach, therefore, is to take the machine apart – remove the disk drives, adapter boards, power supply, and motherboard – and then clean the components individually. (See Chapters 4 and 5 for disassembly and reassembly instructions.)

While the adapter boards are out of their slots, the edge connectors can be cleaned, using electronics cleaning solution, or a hard rubber, artist's eraser. Be sure to blow away any rubber particles that remain on the adapter. Use the wrist grounding strap by connecting it to a grounding point on the board, to prevent electrostatic discharge between your body and the board. Remember that such discharge can easily destroy the chips in a computer.

Lubricating

Lubricating is required only where there is access to moving parts (e.g., keyboard; external disk drive parts such as latches and lever arms; and mechanisms on peripherals, such as cover hinges and paper tray guides on printers). Only the proper lubricants, as listed earlier, should be used. A conductive lubricant is used whenever electrical continuity must be maintained between the surfaces in contact. An example would be in lubricating the grounding strap that rubs against the spindle of a disk drive. Household type lubricating oil should not be used on any computer equipment. Lubricants should be used sparingly, with all excess wiped away to prevent the accumulation of dust.

Adjusting

As with cleaning, attention must be paid to both the environment surrounding the PC and the components of the PC itself.

<u>Adjusting the Work Environment</u>. Heating and cooling should be monitored, with thermostats set to provide the desired temperature for the computers and peripheral devices. Electric fans and electrostatic air filters should not be operated near computers or magnetic storage devices. If humidity is to be increased, use

an evaporative type humidifier; avoid electrostatic, "misting" devices, because of the undesirable electrical charges they generate.

Adjusting PC Interior Components. While the cover is off, check that the interior cables are routed so they are not crimped or twisted; adjust as necessary.

While the motherboard is removed, check that all of the chips are securely seated in their sockets. Chips may creep out of their sockets due to thermal expansion and contraction. If so, they must be pressed back in place. To re-seat a chip on the motherboard, support the underside of the socket with your fingers or thumb, to prevent the board from flexing when pressure is applied to the chip. With the socket thus supported, press the chip down into its socket.

Maintaining Peripheral Devices

Peripheral devices include keyboards, monitors, cables, printers and cables, as well as optional devices installed within the system unit (e.g., disk drives).

Keyboards

Keyboards are easy to maintain. They require only periodic cleaning. If liquid is spilled on the keyboard, the board may be cleaned with a cloth dampened with distilled water. Do not immerse a keyboard in water.

Keyboards should have dust and dirt particles removed at least weekly. Turn the board upside down and shake it gently to remove any loose particles. Turn the board right side up, and remove any clinging dust or dirt by vacuuming with a small (electronics or scale-model vacuum cleaner), or by applying a burst of air from a can of compressed air. Use electronics quality air, available from computer and electronics supply houses. Stubborn particles may first have to be loosened with a soft brush. If the keycaps are dirty, they may be removed for cleaning. Each keycap is individually removable

Note: Do not attempt to remove the space bar on the old PC 83-key board or the AT 84-key board. These space bars are extremely difficult to reassemble.

Pry the cap up with a small screwdriver, or use a keycap extraction tool. Wash the caps with distilled water and a mild detergent. Then rinse and dry. Never use chemicals or solvents on plastic, as the plastic may be dissolved. Professional cleaning solutions designed specifically for electronic components, however, may

253

be used with confidence. Never operate a keyboard or any other electronic device when it is wet. Moisture can conduct electricity and damage the device.

Monitors

Monitors are easy to maintain. They require only periodic cleaning. The display surface of a video monitor has an electrostatic charge that attracts smoke, dust and dirt particles. Clean the glass surface with a soft cloth dampened with distilled water, or use a mixture of isopropyl alcohol and distilled water (1 part alcohol to 10 parts water should suffice) to remove more stubborn grime. Do not use a rough medium such as paper towels, hard-fibre cloth, or any cloth that has elsewhere been exposed to dirt or grit.

Vacuum the surface of the monitor case (or use canned, compressed air) to remove dust. Do not attempt to open a monitor case. Monitors hold a strong electrical charge even when they are not plugged into a wall outlet. The shock could be lethal.

Cables

Cables may become loose, crimped, scraped, cut, or frayed. Signs of wear and tear may indicate problems on the way. If the conditions causing the wear or damage are present, eliminate them. The cable may then last quite a while longer. If the cable is otherwise damaged, be prepared to replace it.

If cables are improperly positioned (crimped, stretched, crumpled), position them to relieve any stress, strain or contortions. Check that the screws or clips that hold the cable connector to the cable, and those that hold the connector to the PC panel, are present and tight. A common source of trouble is carelessness about tightening screws (or positioning the clips) that attach connectors to sockets or panels. Cables and connectors become loose as equipment is moved, especially when the PC is on a stand with wheels and/or pull-out shelves.

Be sure to check all cables and connectors on the outside of the system; internal cables can be checked when cleaning the inside of the unit.

Printers

Impact printers and laser type printers should be vacuumed (or air blasted) regularly to remove dust and paper particles (chaff). Laser printers also may collect toner particles (black powder from within the toner cartridge) that should be removed by vacuuming or spraying with compressed air. For most laser printers, the essential preventive maintenance is done when a toner cartridge is correctly replaced. Please see Chapter 16 for details on replacing toner cartridges.

Disk Drives

Backups

An essential element of preventive maintenance is the regular, systematic backup of data on the hard drives. This is also one of the most neglected areas. There are no valid reasons for failing to backup the drives. There are, however, many "excuses," such as: "I don't know how," "I don't have the time," and "It requires the use of too many floppies." Let's see what backups are all about, and discover how valid these excuses are.

The term "backup" is employed in different ways. In general, a backup is something that can replace another thing, when the first thing is lost or broken. Files can be "backed up" in two ways. One way is to copy them from the hard disk to a removable medium, such as a floppy disk. A file that is copied using the DOS COPY or XCOPY command is immediately usable on another machine. The format of the copied file is exactly the same as it was on the hard disk. Thus, a copied file is a "backup." If the file on the hard disk is destroyed, its copy on the floppy disk can replace it.

Note: XCOPY is far superior to COPY. XCOPY optionally can search for files in subdirectories; ignore files that have not been changed since they were last copied (by XCOPY); prompt you whether or not to copy each file (Y/N); create directories on the target diskette; copy only those files having dates later than a date you specify in the XCOPY command. It has other options as well.

The other type of "backup," such as that provided by using the DOS BACKUP command, also provides a replica of the original file on a floppy disk or other backup storage medium such as tape. Unlike a file that has been copied using DOS COPY or XCOPY, however, a file replica created by using the DOS BACKUP command is not immediately usable on another machine. The file thus backed up has a different format than when it exists on the hard drive. For the file to be used, it must read from the backup disk and written to another disk (usually the hard disk from which it came, or a replacement therefore). This process of reading and writing is performed using the DOS RESTORE command. In essence, the RESTORE command "converts" the file back into the format that is usable on the hard drive (e.g., usable in word processing application).

255

The main advantage of using a "true" backup utility (such as DOS BACKUP) rather than COPY or XCOPY, is that the backup utility can divide a large file across two (or more) disks, and it will prompt you to insert additional disks as needed. COPY and XCOPY do not have this capability. If the disk becomes nearly full and the next file is too large to fit on the disk, the copying process will simply halt.

Depending on one's point of view, a disadvantage of using a backup utility is that it also records the directory structure of the files being backed up. To RESTORE the files to a hard disk, the exact same directory structure must exist on the hard disk. Otherwise, the files cannot be RESTORed.

In summary, copying files to make backups produces files that are immediately usable on another machine. The files may be used from the diskette or copied directly onto another hard drive. It does not matter about the directories on the hard drive from which the files came, or the directories to which they are later copied. A disadvantage is that a file cannot be copied if there is not room for it on the diskette. And, when the copy diskette becomes full, there is no prompt to insert another disk. If there is a large amount of material to be copied, a large number of diskettes may be required.

Bear in mind that it may not be necessary to backup everything on the hard drive. If the original applications (e.g., word processing, data base, spreadsheet) diskettes are available, there is no need to copy them again. All that need be copied are the working files (documents, etc.). For those users limited to 360K floppies, copying files as backups may still be an unwieldy task. The work (and resulting number of disks) is much reduced, however, if the machine has a 1.44Mb floppy drive.

Using DOS BACKUP and RESTORE also may produce a large number of floppy disks. But once started, the backup process is continuous, and you are prompted to insert the disks as needed. To be useable, these backed up files must first be RESTORED to a disk having the same directory structure as the original (source) files. DOS will recreate the directories on the target disk, if necessary. But as a precaution, it may be advisable to print out a directory structure from the hard drive, and store it with the backup disks. To display the directory structure on the screen, type the following command at the DOS prompt. (Use upper-case or lower-case letters; either one will do.)

TREE [enter]

256

To display the directories and the files they contain, enter:

TREE/F [enter]

To print out the directories and files, the commands are:

TREE/F>PRN [enter] or TREE/F>PRN [enter]

(The command characters ">PRN" direct the output to the printer.)

Many users at first find the explanation of the DOS XCOPY, BACKUP and RESTORE commands difficult to understand as they are presented in the DOS manual. The following explanation may be of some help.

The DOS XCOPY Command. The location of files to be copied (or backed up) must be specified in the command. The disk on which the files presently reside is called the "source" disk. The destination of the files to be copied must be specified in the command. This is the "target" disk.

The options that you wish to use with the command are specified by putting a slash (/) at the very end of the entire command (not right after XCOPY) and then a letter after the slash that indicates the option desired. For example, /P says that you wish to be Prompted (Yes or No) whether to copy each file. To use more than one option at a time, another slash and letter is added for each option. For example, /P/E/M/S.

Here are the options:

/A – copies the file only if the file's archive bit has been set (see note below)

/D:mm-dd-yy – copies the file only if the date of the file is the same or later than the date specified as mm-dd-yy. For example, /D:01-01-91 will copy only those files dated as January 1, 1991 or later.

/E – creates subdirectories on the target diskette matching those subdirectories being copied. It will do so even if the subdirectories on the hard drive contain no files.

251

/M – copies modified files, and resets the archive bit (see note below).

/P – prompts for permission to copy each file.

/S – copies the files in the current directory and all files in any subdirectories below the current directory. (See Chapter 7 for explanation of directories and subdirectories.)

/V – verifies (checks size of source file against size of target file – does not compare files character by character).

/W – wait for source disk (this is used if the source disk is a floppy disk)

Note: In DOS, each file has a name and the date it was created or last modified. Each file also has a so-called "archive bit." Recall that a "bit" represents a binary condition, such as "yes or no," "on or off," or "0 or 1." The archive bit for each file can be switched back and forth between two states. In one state, the archive bit essentially says, "Copy me, I have been changed since the last time we did backups." In its other state, the archive bit says, "Don't copy me, I have not been changed since the last time we did backups." When the archive bit is "set," the file will be backed up (see /A, in the above list of options). If the file is copied using the /M option (see /M in above list of options), the /M command "re-sets" the archive bit to its "Don't copy me" state.

The /M option is very useful when using XCOPY to make backup diskettes. If you want to copy a large number of files, and the target diskette becomes full, you can insert a new target diskette, and again command that the same group of files be copied. By using the /M option, the files that have just been copied to previous target disk(s) will be ignored. So, there is no need to keep track of which files are copied or not copied before a target disk becomes full.

The source files (the files to be copied) are specified by their "path name." A path name specifies (1) the drive, (2) the directory, (3) the subdirectory, and finally (4) the file name. (See Chapter 6 for an explanation of path names.)

Suppose we wished to copy all of the files located in a subdirectory called "Work," that was below the subdirectory called "WP51," which was in turn below the root directory of the C: drive.

The source path name thus would be: C:\WP51\work*.* (*.* means "all files" – see Chapter 6 for an explanation of "wild cards").

The target location (usually a floppy drive) is specified in like manner, but much more simply if we are not concerned with directories on the target diskette. The target, therefore, would just be the name of the drive, for example, A:.

Putting all of this together, what would a complete XCOPY command look like? Here is an example. At the DOS prompt, we would enter:

XCOPY C:\WP51\WORK*.* A:/M/P/A [enter]

The command says to copy the files located on the (C:) drive in the subdirectory named "work," which is a subdirectory of "WP51." The "*.*" says, "copy all files." Of course, you could specify just a single file (be sure to include the filename extension).

The command says further that the files are to be copied to the (A:) drive. And, when copying each file, reset its archive bit (/M); ignore any files that already have their archive bit set (/A); and prompt the user for permission to copy each file (/P).

<u>The DOS BACKUP Command</u>. The syntax for the BACKUP and RESTORE commands is similar to that as explained above for XCOPY. It is recommended that the reader examine the XCOPY discussion before proceeding here.

The BACKUP command also has options that are specified by using a slash and a letter as noted above. Details vary from one version of DOS to another; use of BACKUP on versions prior to DOS 3.3 is not recommended. Earlier versions had problems with BACKUP and RESTORE.

Once again the source files and the target must be identified, and the options are again tacked onto the end of the command. Note, however, that the letters designating options for BACKUP do <u>not</u> all mean the same thing as for XCOPY.

Here are the main options:

/S – also backup files existing in subdirectories below the source directory specified in the source path name.

/M – check the archive bit and backup only those files that have been modified since the last time a backup was done.

/A – backup the specified files by ADDING them to the backup diskette. In other words, don't erase any files already on the backup diskette.

259

/D:mm-dd-yy –

backup the file only if the date of the file is the same or later that the date specified as mm-dd-yy. For example, /D:01-01-91 will backup only those files dated as January 1, 1991, or later.

/T:hh:mm –

This is used in conjunction with /D, above. With /T, you can specify the time (hour, minutes) of the files to be backed up, beginning with the date specified by /D. For example, you wish to back up all files created or modified on or after 10:30 am on January 1, 1991, use /T:10:30, and D/:01-01-91.

To use files that have been backed up with the BACKUP command, they must be restored using the RESTORE command, as described below.

The DOS RESTORE Command. The RESTORE command essentially performs the reverse process of creating a backup. In this case, the source is the backup diskette, and the target is the hard drive. For files to be restored, the directories on the backup diskette must be the same as on the hard drive. In other words, the RESTORE program expects to find the same directories on the hard drive that were there when the BACKUP was done. However, RESTORE will create the needed directories if used with the /s option. Note that files can be restored from a backup diskette even if the same (or modified versions of) the files currently exist on the hard drive. Whether or not this will happen depends on the options selected, as explained below.

As with XCOPY and BACKUP, RESTORE has several useful options. Here are the main ones:

/S – restore all files from the backup diskette, including those files backed up from subdirectories.

/P – prompt for permission to restore each file.

/B:mm-dd-yy –

restore only if the file of the same name now on the hard (target) drive has been changed on or before the date specified (mm-dd-yy).

/A:mm-dd-yy –

restore only if the file of the same name now on the hard (target) drive has been changed on or after the date specified (mm-dd-yy).

/M – restore only if the file of the same name now on the hard (target) drive has been changed or deleted since the date the BACKUP was done.

/N – restore if the file is no longer present on the hard drive (target).

/L:hh:mm –

restore if the file of the same name now on the hard (target) drive has been changed at or after the time specified (hh:mm).

/E:hh:mm –

restore if the file of the same name now on the hard (target) drive has been changed at or before the time specified (hh:mm).

CAUTION: Do not use /A, /B and /N at the same time, as this combination would contain conflicting instructions.

The DOS commands XCOPY, BACKUP and RESTORE have been covered in some detail here because they are provided as part of DOS itself, and are thus readily available. Many experts in the computer field are not favorably disposed toward the DOS BACKUP program. In part this may be due to the fact that prior to DOS 3.3, the program contained several bugs, and was not considered reliable enough for something as important as backing up a system.

Also, pre-DOS 4.0 versions of DOS BACKUP permitted the hidden system files on the hard drive to be backed up along with the other files. If the BACKUP was done and a later version of DOS was then installed on the machine, a problem occurred during RESTORE: the old system files on the backup would be written over the newly installed system files on the hard drive. With later versions of DOS, however, this is not a problem.

Finally, there are numerous third-party backup programs available, which are easier to use (more "user friendly") than DOS BACKUP and RESTORE. Two examples are:

- Fastback – by 5th Generation Systems
- DS Backup – by Design Software

As an alternative to floppy disks, there are many "hardware' options, including tape systems, for backing up drives. These systems are beyond the scope of this book. Not to be overlooked, however, is the possibility of installing an extra hard drive and using it just for backups.

Now that the system is safely backed up, we may return to the more mundane tasks of preventive maintenance.

Cleaning Disk Drives

The surfaces (housings and exterior surfaces) of disk drives are cleaned the same as are other components inside the unit, as described further on. Since the hard drive head assembly is sealed, the heads are not to be cleaned.

On a floppy drive, the read/write heads accumulate a coating of iron oxide from being in contact with the floppy disks. If allowed to persist, this coating will cause the heads to malfunction.

Floppy drive heads may be cleaned by using a special, cleaning diskette onto which a small amount of head cleaning solution is placed. The dampened diskette is simply placed into the drive, and the drive is caused to spin by entering a DOS command at the keyboard. When the drive spins, the drive heads contact the fabric containing the cleaning solution, and the iron oxide deposit is transferred from the heads to the diskette. The diskette referred to here is called a "wet" diskette, and it is to be preferred over other types of cleaning diskettes. There are "dry" or "hard" diskettes that attempt to clean the heads by abrasion, but these are not recommended.

To spin the diskesst for cleaning, any command that requires the diskette to be read will do. For example, the DIR command will produce an attempt to read the diskette. An error message will be displayed, since the cleaning diskette is not readable. By repeating the command several times, the disk should be caused to spin enough times to clean the heads.

Note: After cleaning the heads with a wet diskette (or using any liquid solution applied by any other means), allow time for the heads to dry. If a working (e.g., program or data) diskette comes in contact with the heads while they are still wet, the data diskette will be damaged.

262

Adjusting Disk Drives

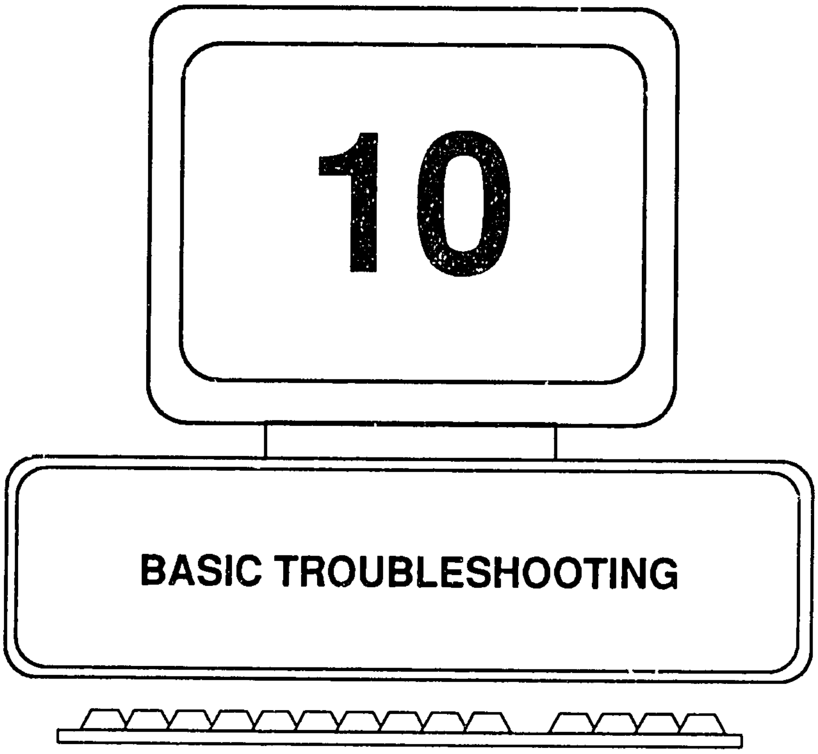
Alignment. Any drive operated by a stepping motor will eventually lose some of its positional (track and cylinder seeking) accuracy. The loss of accuracy may be foretold by an increasing number of read/write errors. However, there are compensating functions in both the disk controller and DOS that will mask the progress of this condition. Specifically, the error may be "corrected" by the machine automatically reading the sector again, unbeknownst to the PC user. There are diagnostic packages available that are used in conjunction with special diskettes to align the heads, but they are beyond the scope of the present book.

Reformatting. As noted in Chapter 6, the surfaces of hard drives that store data are first prepared by low level formatting. This process writes information magnetically onto the surfaces of the cylinders (like a kind of map) in order to later control where data is to be written. Over time, this magnetic information fades, and the "map" becomes unreliable. The information can be restored by again performing a low level format. Special software is available for doing a "non-destructive," low level format on a drive that contains data. This procedure reads the data from the disk, stores it in memory, formats the disk, and then writes the stored data back to the disk. Nonetheless, the disk should be backed up beforehand: if the power fails during the formatting process, the data in memory will be lost. A backup is essential!

A "destructive" low level format can also be done. But, as its name suggests, the data on the disk would be destroyed. The details of these procedures are beyond our present scope. The reader should be aware, however, that low level re-formatting is considered an important part of preventive maintenance. It is also important to know that a hard disk that has "failed" (other than by physically crashing) can often be resurrected just by doing a non-destructive, low level format.

The Squealing Hard Drive. Squealing sounds in mechanical devices are caused by friction and vibration – due to one surface rubbing against another. A common source of squealing in hard drives is the friction and vibration that occurs between the drive motor spindle and a static charge grounding strap that rests against the spindle as the spindle turns. The strap is on the bottom of the drive. To see it, it may be necessary to remove the circuit board on the bottom of the drive.

Although the noise made by this condition sounds serious, it may be cured (usually) by placing a small glob of silicon caulking on the outer surface of the strap (not between the strap and the spindle). The caulking adds mass to the strap and thus changes its vibration frequency. Normally, this is sufficient to eliminate the noise. Even if the caulking falls off after a few months, this "fix" is better than unnecessarily replacing the hard drive.


10
BASIC TROUBLESHOOTING

CHAPTER 10: BASIC TROUBLESHOOTING

Troubleshooting is the art of finding out what is wrong with a PC and then correcting the problem. By this definition, any steps taken to identify the cause of a malfunction and to restore the system to good working order might be considered "troubleshooting." Certain problems, however, such as a loose connector, a frayed cable, or an accidentally erased program, form a class of relatively simple events in comparison to those that may require the use of a diagnostic software package in finding a remedy. Accordingly, we devote the present chapter to the less complex, essentially mechanistic problems, and reserve the treatment of diagnostics packages for the following chapter. In the present chapter, we examine the importance of establishing a rational, systematic approach to diagnosing problems; the need for preparation before problems arise; the critical importance of (and procedures for) backing up a system before troubleshooting it; and some basic procedures for identifying the causes of malfunctions when they ultimately do occur.

Develop a Rational Philosophy of Troubleshooting

In most of our life's experience, when a machine fails to operate correctly the problem is usually electrical or mechanical. The natural reaction is to reach for the tool box or call in a mechanic. It is only human nature to transfer the experience to a computer when it begins to malfunction. Unfortunately, that approach is normally a mistake. Because so much of what a computer does depends on what its operator does (or fails to do), many problems begin and end with the "human" component of the system. In an otherwise healthy machine, problems may also arise due to errors (bugs) in the applications software or the operating system.

The list continues with problems due to conflicts between different files (programs) which, on their own, would run correctly. In addition, whenever the system is set up (configured) incorrectly, or the configuration information has been disturbed, performance is degraded accordingly. Then there are transient problems caused by fluctuations in the power supply. None of these problems will be solved by taking the PC apart. And, unless the problem clearly and obviously lies with a mechanical or electronic component, most of the foregoing possibilities should be systematically eliminated before reaching for a screwdriver.

Another quite natural expectation is that any machine would have a finite number of possible problems and solutions, just as a game of chess has a finite number of possible moves. Therefore, why not simply list or diagram all possible events along with their solution. Then when a problem arises, the solution could be looked up in a book. The answer is that there are just too many possibilities. The list or diagram would be so large and so complex, no one would be able to use it. Furthermore, the author of the list would need to know ahead of time all possible uses of the machine; the details of all possible software programs to be run on it; and the characteristics of all possible peripheral devices, cables and adapters that might be installed. This would be an impossible task.

There are reference works that list many of the most common symptoms, probable causes and possible solutions. The references provided at the back of this book are a few good examples. There are also the indispensable tables that explain what error codes mean. These are available from the manufacturers and vendors for each particular machine. Error codes (and messages) are those cryptic numbers (and words) that appear on the monitor (or those mysterious beeps emitted from the speaker) when the PC finds something wrong with itself.
For a vast number of additional problems, however, there are unfortunately no error codes to be seen. Success here depends largely on the general troubleshooting approach taken, and on experience that comes only with time.

Effective troubleshooting is based on a logical sequence of questions and answers, classifying symptoms, and isolating faults. The general approach does indeed resemble the diagram mentioned above, but it is far less detailed and complex. And, it may exist not on paper but in the mind of the troubleshooter. However humble or simple that mental diagram may be, it is far superior to any form of trial-and-error, hit-or-miss approach.

If all possible problems and the paths to their solution could be depicted as a chart, the chart might resemble a gigantic, upside down tree. Starting at the top and working downward, each symptom, condition and test would branch off to yet another series of paths and forks in the tree. Eventually (and ideally) you would arrive at the end of a specific branch that revealed the exact cure.

With such a diagram in hand, you could begin at the top and laboriously work your way through to the solution. But ordinarily, you would make a preliminary diagnosis, and on that basis you would "jump into the tree," perhaps somewhere in the middle, and pursue the problem from there. If you enter the tree in the wrong place, you can waste a good deal of time.

For example, if you incorrectly decide that a hardware component is the culprit, you bypass many of the earlier branches in the troubleshooting tree, one of which is the correct solution. You cannot return to that point by now taking the machine apart. So, the more you can find out ahead of time about the PCs under your care, the more reliable your initial diagnosis will be. Your success will depend not only on your skill with a screwdriver or voltmeter, but on the planning and preparation accomplished long before a problem occurs.

In the remainder of this chapter, an approach is suggested to help organize your thinking. Also, it will prepare you to understand more quickly the material found in more advanced texts. With experience, you will no doubt develop your own basic approach and tailor it to your particular working environment.

Prepare Before Problems Occur

Basic troubleshooting begins with effective preparation before problems arise. Having the proper records, information and system of data collection in place will greatly reduce the time and effort needed to solve problems when they finally do occur.

Obtain Baseline Data

Whether you are responsible for maintaining the PCs of others or just your own machine, one fact remains: the more you know about a machine before it has a problem the faster you will be able to solve the problem. Most problems occur because something "different" has happened. True, a component may fail due to old age or stress, and taking the machine apart to replace it may be necessary. But that possibility should be far down on your list of basic troubleshooting steps. There is much to be done before arriving at that point. Even before the need for a troubleshooting session and possible repairs, important groundwork must be laid.

Just as every good doctor needs a record of the patient's medical history and vital signs as a first step, you should have a record that describes the PC's make, model and installed options; the data needed to configure it; and a record of how well it was performing before the complaint appeared. Thus prepared, you can focus your investigation and measure the extent of the problem (and the effectiveness of your remedy).

How, then, does one go about getting that all important "medical history"? One easy way is to run any of several commercially available programs ("diagnostics") and save the results, either on diskette, on a printout, or preferably both. Many think of diagnostics as something to be used only after a problem occurs. But by then the machine may be too "sick" to run the diagnostic program. If so, you may lack important information needed to restore it quickly to health. In addition, by running the diagnostic package on many different, healthy machines, you gain experience in using the package without the need to remedy malfunctions at the same time.

For illustration, we have selected the diagnostic package, "Checkit," the use of which is described in the next chapter. In brief, this diagnostic package allows you to see how the system is configured, and it tests all of the main components including memory, disk drives, display, communications ports, and keyboard). It displays the results of each test, and allows you to print them out or to save them to a disk file (which, of course, may be printed later.) The package is fully menu-driven and simple to use. The first step is to obtain the PC baseline information and store it in a safe, accessible place.

Collect and Organize Documentation

Collect as much documentation as possible (including any diagnostic and setup diskettes that come with the PC) and have it well organized.

Prepare Your Own Diagnostics Kit

Have your own diagnostics kit well stocked and organized. This includes boot diskettes for the different versions of DOS you may encounter, and legitimate copies of third party software such as Checkit, Norton Utilities, and as many others as you can afford. Each of these tools has a different profile of capabilities.

Conduct Preventive Maintenance

Institute a preventive maintenance program (see Chapter 9 for suggestions). The best maintenance program in the world can be defined in just three words:

"BACKUP, BACKUP, BACKUP!"

What? Is backing up the hard disk really "maintenance?" Think about it. If a hard disk physically crashes, you remove it and replace it. Certainly, this is maintenance. But, if you cannot then restore the data that resided on the failed disk, the "maintenance" is incomplete and unsatisfactory. You have not solved the problem, because you have not restored to system to its pre-problem condition.

Considering the risks involved in troubleshooting a PC, anyone responsible for providing technical support to PC users may well consider establishing the following rules:

- Support will be provided only to those users who demonstrate responsibility by performing regular backups of their hard drive.
- Troubleshooting will not be performed on a machine unless the hard drive has been backed up as recently as possible.

If these rules seem overly strict, consider this: what if you start out to fix an apparently minor problem, and later discover that data on the hard drive has been lost. Maybe you caused the problem, or perhaps you did not. But whom do you think will be blamed for the loss? If you are to be the "PC Doctor," the backups are your "malpractice insurance." Components are relatively inexpensive and easy to replace. Data is expensive, and there may be no way to replace it, ever!

Log Trouble and Service-Calls

Establish a trouble-call reporting procedure and troubleshooting/service log. The same problems may recur on a particular machine or across several machines. By keeping a data base of the problems encountered and their solutions, you will develop an invaluable cache of information that will save time, money and frustration. You will learn more quickly from experience, and you will avoid duplicating previous efforts whenever a common problem reoccurs. But if the data collection forms and data base are not in place ahead of time, all of this valuable information will escape.

When a Problem Occurs

Despite all best efforts, problems will occur. If you have performed the steps as outlined above, however, the problems will be less frequent and less severe. Further, you will be able to address with much greater efficiency and success those problems that do occur.

Conduct a Thorough, Preliminary Inquiry

A common mistake among novice troubleshooters is to attempt a cure prematurely. Often, this unnecessary haste is fostered by the PC user, the victim of the malfunction, who desperately needs to have the machine up and running again as fast as possible. "Quick, fix it," is the cry. "I need the report out by noon!" Your compassion is in order here, but not at the expense of working thoroughly and methodically. Mere action is not a substitute for accurate work.

Being fully prepared as suggested earlier, you will have in hand the correct documentation, service record and configuration data for the PC to be worked on. And, the hard drive will have been backed up. The type and order of the questions that you ask when you arrive at the scene of the calamity will undoubtedly vary from one situation to another. But, avoid the trap of narrowing your search for a solution too quickly. Further, if you are unable to resolve the problem, it will be important for you to have an accurate description of the machine and its symptoms, so that this information can be reported to the vendor or other source of outside support. Most of the job is in being able to define the problem correctly! The following questions will help to focus your investigation.

- What is the PC doing now, exactly? Is it dead or will it perform certain functions but not others? Will it complete the boot sequence? Is it displaying an error code? Is it making strange noises?

- Has the problem occurred before? If so, how was it solved? Do you have the problem and its solution logged into your data base of common problems?

- Did the problem appear gradually, or all of a sudden? For example, you may receive the following report: "The disk drive has been making a screeching sound off and on for about a month. Until now, banging the side of the PC [Good Grief!] has made it go away. But now, the screech is louder, and banging on the PC won't stop it." Or perhaps, the report is, "The ceiling lights flickered and then my PC just froze up."

- What exactly did the user do just before the problem occurred? What is needed is not just a general statement to the effect that, "I was running Word Perfect, or Lotus 1-2-3, and the machine froze up." If possible, it should describe the operation or function being attempted when the problem occurred. For example, "I was running a new macro and the machine just hung up."

- Who else uses the machine and when did they use it last? Maybe they came in for a short work session and meanwhile changed something on the machine, and failed to let anyone else know about it.

- What did the user attempt after the problem occurred? Press keys at random? Try to reboot the system? Check the power outlet and cable connections? What? It is unlikely that the user did nothing at all. Neither is it safe to assume that whatever the user tried or checked need not be tried or checked again by you. When the user exclaims, "I already tried that, why are you wasting time doing it again?" just be patient and quietly continue your usual, systematic approach.

- Is everything plugged in, and is everything plugged into the correct place, with the connectors in the correct orientation (not backwards or upside down)?

- Are the cables and connectors in good condition, or are they bent, frayed, crimped, or cut? Note that not all of these conditions are easily visible, if they can be seen at all.

- If the machine appears dead (no display, no response to the keyboard), are there any signs of life at all? Can the fan motor be heard running? Do the disk drive lamps light up. Are the keyboard lights working?

- Is there a proper source of electricity? Does the wall outlet have power, or was the power turned off by a switch on the wall somewhere in the room or out in the hall?

- Do problems seem to occur at a particular time of day or day of the week? Perhaps everyone in the building is making coffee or running equipment that draws a lot of electricity and thus causes fluctuations in the power supplied to the PCs.

- Does the machine have virus protection? Has any software from a bulletin board or "community" diskette been loaded?

- What is new or different about the PC or its environment? Has a different printer been connected to it? Was a new adapter card or some memory installed? Was some new software put into use? Were some features of an application used that were not used before?

- Was the machine moved for any reason? For example, was it moved a few feet when the cleaning crew shampooed the carpet; was it moved across the room to a new wall outlet, or was it moved down the hall to another location altogether?

- Has the environment changed? Maybe the air conditioning was turned off over the weekend, and on Monday morning the PC refused to boot up. When was the last time the machine operated properly?

These kinds of questions should provide a key to the steps to take next. If the cause of the problem is not evident by now, proceed as outlined below.

Do the Obvious and Easiest Things First

Avoid the trap of assuming that things are worse than they really are. Much time can be wasted looking for a nonexistent, difficult problem simply because something obvious was overlooked or assumed to be all right. Take nothing for granted!

- Check the power source – use a tester or plug a lamp into the wall outlet.

- Check external cables and connectors; unplug and reconnect them. Don't forget to check that all plug-in items are properly seated, including any font cartridges used by the printer. Often this will solve an invisible problem.

271

- Be sure that everything is plugged into the right place. Certain items such as telephone lines for the modem and the auxiliary telephone instrument are not key-coded or shape-coded. It is easy to plug them into the wrong jack.

- Turn the machine off, wait to a slow count of five, and reboot. If this fails, try booting from the A drive, using one of the boot diskettes from your "tool kit" that you prepared ahead of time. Be sure it contains the proper version of DOS. A problem arises commonly when the user has somehow erased the COMMAND.COM file. If so, the machine will boot up part way, and then display the message:

  "Missing command interpreter"

  To solve this problem, simply boot the machine from the (A:) drive, and copy COMMAND.COM from the (A:) drive into the root directory of the PC hard drive. Assuming the hard drive is C:, and the A> prompt is on the screen after booting from the floppy, the command is:

  Copy COMMAND.COM c:\ [enter]

  Then remove the floppy disk from the (A:) drive (remembering to leave the floppy drive door open or unlatched), and reboot the machine.

CAUTION: Do not install COMMAND.COM from an IBM diagnostics diskette for use on the hard (boot/default) drive. It is not a complete file, and it is not intended for operating the PC in normal use.

- Check the condition and status of each peripheral. It is not uncommon, for example, for a PC to be hung up simply because the printer has gone off line, overloaded its buffer (for example, in trying to process a large graphic), run out of paper, or had the paper tray jarred loose. Turn the printer off, wait 10 seconds and turn it on again. Be sure the printer "on-line" lamp is lit. Be aware also that certain applications will hang the PC if they try to print to a non-existent printer, or to a printer different from the one they expect.

- Try to reproduce the problem on another machine, especially if you suspect that the problem lies in an application package. Some of the more advanced word processing functions, such as working with column formats, tables, merges and macros in the same document, can cause a machine to hang. These problems are not limited to a specific PC.

272

- Try to reproduce the problem on the faulty machine. Be sure everything is backed up before attempting to induce a fault! Perhaps rebooting the PC has cleared the problem temporarily but not cured it. Write down the steps and keystrokes you use. Then, when the problem recurs, you will know what actions and events produced it.

- Isolate the fault by separating the elements of the system, first the software and then the hardware. You are following a process of elimination. Logically, you tackle the most likely suspects first. Software is to be suspected before hardware.

Isolate the Software

If the problem occurred during a work session using a particular application, as mentioned earlier, the problem may be isolated from the malfunctioning PC by running the application on another machine and attempting to duplicate the problem. A different copy (but the same release date and version) of the application may yield an answer. But, if possible, it is best to copy the application software off of the problem machine and run it on a healthy machine. The copy on the problem machine may have become corrupted.

Certain applications (or a newly installed version of an operating system) may need more memory (RAM) than is available on the PC. They may produce the familiar error message, "Insufficient memory." Be aware that each successive version of DOS has increased in size. An application that ran well under DOS 3.0 may not run under DOS 5.0 if your supply of memory was marginal to begin with. Unless DOS is loaded into upper memory, it consumes memory within the 640K of "user" memory (RAM). Once loaded, there may not be enough of the 640K remaining to load the application and/or create any sizable file with it. Also, adding options (and thus driver files), TSR programs, menu utilities, memory caches, RAM disks, and anything else that competes for RAM may produce a memory shortage.

Applications that automatically change settings such as the number of files and buffers, display parameters, or printer instructions may cause a problem. All may be well while the application is in use, but some of these applications terminate without restoring the changed settings to their original values. Then, in using another application, you may receive a message such as, "Too many files open;" or see your color monitor now displaying only black and white; or observe your printer mysteriously generating documents in the wrong font or orientation.

If you have done what you can to rule out a problem with the application software, proceed as outlined below.

212

- Boot the PC from a floppy that has the same version of DOS as that normally used on the PC in question. The boot diskette should not contain a CONFIG.SYS file or an AUTOEXEC.BAT file. If these files are on the hard disk (and they usually are) they may have become corrupted. Or, they may be loading programs and drivers that are causing memory management or other resource conflicts. By booting the system from the "virgin" floppy, these files will not be loaded. If the PC then boots and runs correctly (not all devices may operate, however – see below), examine the AUTOEXEC.BAT AND CONFIG.SYS files on the subject PC's hard drive.

  Note that peripheral devices require special programs called device drivers to be loaded when the PC is booted up. If the driver is absent (not loaded by the CONFIG.SYS file), or if the file is the incorrect version or has become corrupted, the peripheral device will not run. This applies also to certain plug-in font cartridges A particular driver file for the cartridge must be available on the hard drive, and the machine must be able to find the drive and directory in which the file resides.

- Drivers loaded by CONFIG.SYS may also produce conflicts in the use of interrupt request (IRQ) channels or direct memory access (DMA) channels, because of incorrect system configuration. For example, two devices may be trying to use the same IRQ channel. This is not allowed, and the system will freeze up or show other undesirable symptoms. Or, the PC may fail to boot up altogether.

- Be especially alert where terminate-and-stay-resident (TSR) programs such as screen blankers and pop-up utilities (scheduler, phone list, calculator) are installed. These programs remain in memory while the machine is powered on, even when they are not actually in use. It is common for them to conflict with other demands placed on memory during normal operations. As a result, the machine freezes up.

- Run a diagnostic program such as Checkit (see Chapter 11). The value of doing so at this point, of course, depends on your diagnosis thus far. You may have decided to run it sooner, or not at all. The program, however, may reveal certain problems, especially those concerning the system setup (configuration) and the present performance of the major components. If the configuration is not correct, the related components will not function at all, or they may operate incorrectly. If you obtained the configuration data ahead of time, as recommended earlier, you can now run the setup program and key in the needed data. If you failed to run the diagnostic ahead of time, the machine may be unable to run it when a problem occurs.

If a hardware problem such as a failed power supply is not obvious in the first place, or if an error code does not present itself as the possible problem, eliminate the following possibilities next:

- A corrupted power source (e.g., undervoltage, overvoltage, surges, spikes)
- Loose connections, faulty cables
- Peripherals (e.g., printer causing PC hang up)
- User error
- Applications errors (bugs)
- Terminate-and-Stay-Resident (TSR) memory conflicts
- Missing, incorrect, or corrupted drivers
- Improper configuration

Then, you may wish to turn your attention to the hardware. Once again, do the obvious and easiest things first. Remember: divide and conquer – separate the components to isolate the fault.

Isolate the Hardware

If the problem is due to a hardware component (this includes all electromechanical and electronic components), the defective component may be interacting with other components, thus producing symptoms that are difficult to analyze. Insofar as possible, the strategy is to test each function and component in isolation. As a simple example, let's say the keyboard is not functioning. Try a different (identical or compatible) keyboard. If it works, the problem is either in the original keyboard or its cable or plug. Alternatively, take the original keyboard to another PC and try it there.

The same strategy applies to all of the peripherals, such as monitors and printers. But take care! Never swap components until you are certain that they are compatible. Outward appearances are deceiving. Also note the if you move a display or a printer, for example, to another PC, be sure the PC is configured properly to use the peripheral. Remember that the PC has no way of knowing that you have changed its equipment. Further, remember to restore the configuration on the healthy machine after you are finished testing the swapped item. Once again, you can see the value of having the current configuration data available for all PCs under your care.

A variation on the above approach is as follows. Disconnect all peripherals, and then reboot the machine. Reconnect each peripheral, one at a time, and reboot the machine after connecting each one. Peripherals usually have three main components: an adapter card, the peripheral device itself (monitor, printer, keyboard, disk drive) and a cable. Check (or swap) the components in the easiest order. This means the cable first, then the device, and finally the adapter card.

What has been addressed thus far is by no means exhaustive. But it does illustrate in general the required systematic approach. And it emphasizes that there is usually much to be done before deciding to open up the machine. Assuming we have not yet discovered the source of the problem, let's now look inside the machine.

CAUTION:

- **Never open the case of a power supply. It contains a highly dangerous electrical charge, even when it is not plugged in.**
- **Never operate a power supply whose output is not connected to the PC. The power supply may explode!**
- **Never attempt to open up a display monitor. It contains a highly dangerous electrical charge, even when it is not plugged in.**

Allow the PC to warm up for about 20 minutes, to reach it normal operating temperature inside the cover. Then turn the PC off, unplug its AC power cord, and remove the cover. Remember that all of the precautions and procedures for PC disassembly and reassembly, as outlined in Chapters 4 and 5, apply just as well to troubleshooting.

Immediately after removing the cover, touch the CPU, RAM and ROM chips to see whether they are operating at the correct surface temperature. The chips should feel warm but not hot. A component that feels either excessively warm or cool should be suspect. Depending on the particular machine you are working on, you may or may not have immediate access to the CPU, RAM and ROM chips without removing other components first.

Press down on the socketed chips (CPU, RAM, ROM) to ensure they are seated and have not crept out of contact due to thermal expansion cycling. Check that the SIMMS are properly installed and seated.

Check the condition and installation of all internal cables and connectors.

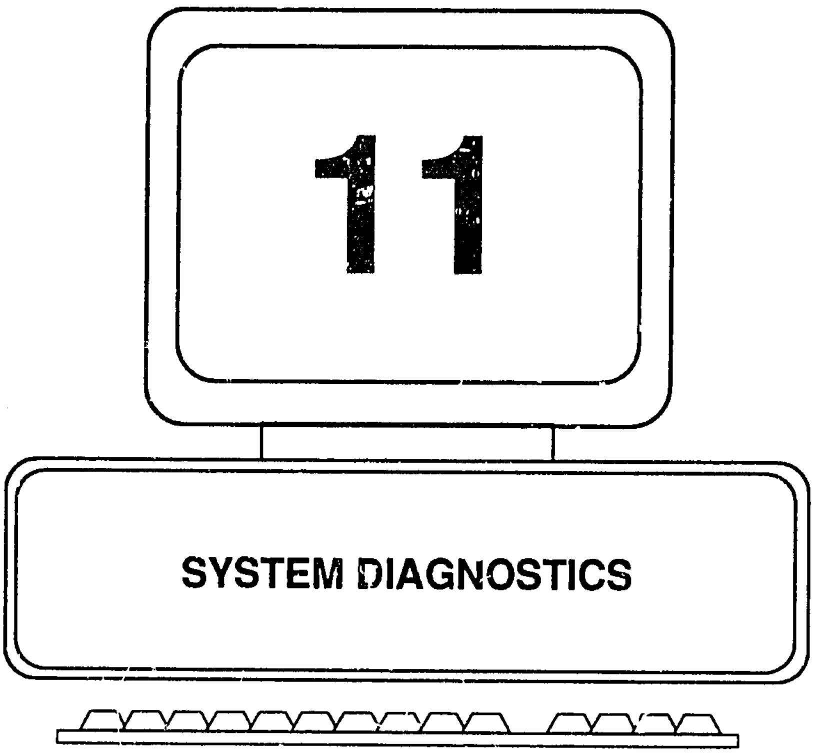
Check that the adapter cards are properly seated in their slots; the cables are properly attached; and the configuration settings are correct. DIP switch and jumper settings, however, need not be suspect unless someone has been working on the inside of the PC since the time that it last operated properly.

Sometimes just removing and reinserting an adapter card will clear a problem. Sometimes moving an adaptor to a different slot will clear a problem. (Remember that you may need to reconfigure the system if you move an adapter.) Take care not to disturb the configuration settings on the adapter card.

Avoid touching the edge connectors and other electrical contact points of components. The oil and acid from your skin is harmful to them. Contacts can be cleaned with ordinary rubbing (isopropyl) alcohol and a soft cloth. Do not use loose, fibrous material such facial tissue paper or ordinary cotton swabs. The fibers come off and remain as debris on the connector and elsewhere in the system.

Be sure that the cooling fan is powerful enough and that nothing (either inside or outside the PC) is blocking the air flow through the system. Intermittent problems, those that come and go for no apparent reason, may be due to thermal stress. When objects are heated they expand; when the are cooled, they shrink. Continued expansion and contraction can physically damage a component such that it works only part of the time, works unreliably, or works not at all. To diagnose thermally related problems, it may be necessary to observe the machine over an extended period which duplicates the temperature cycle in which the problems occur.

NOTES


11
SYSTEM DIAGNOSTICS

CHAPTER 11: SYSTEM DIAGNOSTICS

Categories of System Diagnostics

Compared to "troubleshooting" in general (which includes checking mechanical and electrical connections, visual inspection, and so on), the term "diagnostics" typically refers to tests built into the machine itself or software designed to evaluate the status and performance of the system. Overall, diagnostics may be said to fall into the following four main categories:

1. Power-On Self-Test (POST)
2. Routine Diagnostics
3. Advanced Diagnostics (IBM)
4. Third-party (Aftermarket) Diagnostics

Power-On Self-Test (POST)

When a PC is powered up, it automatically performs a brief test on itself. It checks that memory is functioning and that certain major hardware components are installed/operating. The instructions for this test are stored in the ROM BIOS on the machine. The test bears the logical name: Power-On Self-Test, or "POST." When referring to the test instructions listed in the BIOS, the test is referred to as the "Power-On Diagnostics," or "POD."

Assuming the PC is operating properly, the POST runs when you turn on the PC. The POST is what displays the sequence of numbers on the screen as it checks the installed memory. The BIOS version may be displayed briefly on the screen; the speaker may emit a single short beep, and the NUM lock lamp will be lit on the keyboard. What actually happens and what is displayed or heard, however, depends on the PC in question. It is not important for you to memorize the details of what the POST does. If the POST detects a problem it will report it with an audio signal (beep) or an error code on the screen. In general, however, the POST checks the following:

1. Microprocessor

2. System memory and buses

3. Primary display (any secondary display installed is not tested)

4. Keyboard (interface, and whether any keys are stuck)

5. Presence of disk drive (and status, if present)

6. Number of ports installed

On Micro Channel Architecture (MCA) systems, specifically PS/2 Models 50 through 8 , additional test and capabilities are provided. Among them, the Programmable Option Selector (POS) will automatically detect and configure any installed adapter cards.

On PS/2 systems, the POST is referred to simply as an "internal self-test." If the POST or its variant, depending on the system, is successful, you will hear one short beep. Note that the single short beep may occur in the presence of a blank screen, or the PC may fail to boot, so there may still be a problem.

If a problem is detected, you may hear any of several "beep codes," or see an error code number on the screen (assuming, of course, the display is working). Figure 11-1 lists the audio signal error codes for IBM machines. (Clones may differ from the signal definitions shown).

Fig. 11-1. IBM Audio Error Signals (POST)

Signal: • = short beep — = long beep	Display shows	What May be Wrong
None	none	no power or bad supply
None	cursor only	power
None	DOS prompt	speaker
•	DOS prompt	normal
•	PC BASIC screen	floppy disk/drive/adapter
•	no boot	floppy disk/drive/adapter
•—	none	monitor cable/adapter
••	wrong or blank	monitor cable/adapter
••	an error code number	see Fig. 11-2
repeated •	Error code 305	keyboard fuse
repeated •	Anything other than the above	power
continuous tone		power
—•		system board
—••		monitor
—•••		monitor

Figure 11-2 presents a partial listing of major error code categories for IBM PCs. Note that codes may vary from one model of machine to another, and that the codes shown here do not apply to MCA systems. Also, it may require use of advanced diagnostics software for certain codes to be displayed. The first two digits of each code indicate the area or component of the system where an error has been detected. The "xx" following the first two digits is replaced by specific

281

digits when the error code is displayed. Each "x" may have a value from 0 - 9. Not all combinations of numbers are used, however.

Fig. 11-2. Examples of Error Code Major Categories

Error Code	Problem Area
01xx	System board
02xx	Memory
03xx	Keyboard
04xx	Monochrome display
05xx	Color/Graphics display
06xx	Diskette drive/adapter
07xx	Math coprocessor
09xx	Parallel printer adapter
10xx	Alternate printer adapter
11xx	Asynch comm device(s)
12xx	Alternate comm device(s)
13xx	Game control adapter
13xx	Color/graphics printer
15xx	SDLC comm adapter
16xx	Display station emulation adapter
17xx	Fixed/hard drive disk or adapter
18xx	Expansion unit (PC/XT)
20xx	Bisynch comm adapter (BSC)
21xx	Alternate BSC
22xx	Cluster adapter
23xx	Plasma monitor adapter
24xx	EGA adapter, or MCA video
25xx	Alternate EGA adapter
26xx	PC/370-M adapter
27xx	PC/3277 emulation adapter
28xx	3278/79 Emulator, 3270 con. adapter
29xx	Color/graphics printer
30xx	LAN adapter
31xx	Alternate LAN adapter
32xx	PC display adapter
33xx	Compact printer (PC/XT)
39xx	Prof. Graphics Controller Adapter
48xx	Internal modem
70xx	Chip set (Phoenix BIOS)
71xx	Voice comm adapter
73xx	3.5 in. diskette drive
85xx	2Mb memory adapter
86xx	PS/2 pointing device/mouse
112xx	SCSI adapter
194xx	Adapter memory module
210xx	SCSI fixed disk controller

Routine Diagnostics

On the IBM family of PCs, the POST is supplemented with a collection of procedures and utility programs to assist in verifying and/or changing (AT and PS/2) the system configuration, preparing diskettes for diagnostic use, copying diagnostic diskettes, and parking the heads of the hard drive in preparation for moving the machine.

For the PC, XT and AT, the diagnostics diskette is provided along with the Guide to Operations Manual that accompanies each machine when purchased. For the PS/2 series of PCs, the programs are found on the "starter" diskette or "reference" diskette provided with each machine.

Caution: Diagnostics diskettes must be used only on the model of PC they were written for. Check that the disk you intend to use is for the PC and XT, or AT. There is a version available for use on both the XT and AT, 286 machine. Using the wrong diskette will produce invalid test results. Using the wrong diskette to park the hard drive heads may result in loss of data from the hard drive. SHIPDISK should be run only from the diagnostic/reference diskette, not after copying it to the hard drive.

Note that when diagnostic/reference diskettes are issued, they are limited to the PC optional devices available at that time. As new devices are introduced and a changes are made to the diagnostics programs, the new programs are distributed on floppy disks. To create a single diagnostic diskette containing the appropriate and up-to-date files, it is necessary to copy/merge the files from the several diskettes onto your single diskette by following the steps accompanying the update diskettes. However, if you acquire a non-IBM device that has its own separate diagnostic diskette, the diskette should be used by itself and not be merged with your other (IBM) diagnostic files.

To use a diagnostic diskette, you must boot up the system with the diskette in the A drive. Insert the diskette and switch on the machine. Alternatively, if the PC is already powered up, you can simple perform a "warm boot." Hold down the CTRL, ALT and DEL keys at the same time. Then release them. The PC will boot from the A drive and will typically display a menu. From there, the choices are essentially self-explanatory. You can check out the system (run diagnostics routines), format a diskette, copy a diskette, prepare the system for moving (park the hard drive heads), run the setup utility to configure the system, and set the system date and time. You will encounter minor variations in menu format and options, depending on the age and model of your PC.

283

Caution: The format option on the diagnostics disk is for preparing a special "scratch" diskette to be used in testing the system. A diskette formatted by the DOS FORMAT command cannot be used in its place. And, the diagnostic format option should not be used to format diskettes for general use.

Advanced Diagnostics

It is unlikely that the average user or the beginning level support person would need to venture into the area of advanced diagnostics. The material is voluminous, and the repairs are probably beyond reach. Further, there would be the risk of causing problems or destroying data, for example by running a low level format incorrectly.

As a matter of general information, IBM sells its Hardware Maintenance Service manual for about $200. Diagnostic procedures and diskettes are included. The manual is intended mainly for qualified technicians. It covers the PC, XT and AT. Supplements to it cover the XT-286 and the PS/2 Models 25 and 30. The Model 25 and 30 PCs, of course, are ISA machines, rather than MCA systems.

A separate manual (for around $180) is sold for the MCA based, PS/2 machines (Models 50-80), which also includes a set of diagnostic diskettes – but wait!. If you have an MCA PS/2, the diagnostics are already on the reference diskette that came with the machine. Simply boot up from the reference diskette. When the main menu appears, hold down the CTRL key and press A. The menu for advanced diagnostics will appear.

The advanced diagnostics, whether for the PC, ISA or MCA architecture are much more rigorous, comprehensive and detailed than those found on the "user level" reference/diagnostic diskettes. In many cases, they are able to diagnose the malfunction right down to a specific part. The manual then provides information on adjustment, maintenance and removal/replacement. It also includes part numbers essential for ordering replacement parts or spares.

IBM's advanced diagnostics are quite good, but they are limited in their testing of disk drives. Disk drives are prone to failure (relative to other parts of the system). And, unlike other components that can be replaced at will, the hard drive contains precious data to be recovered, salvaged or restored in the event of a drive malfunction. Fortunately, there are numerous alternatives (or supplements) to the diagnostics provided by IBM.

Third-Party (Aftermarket) Diagnostics

As its name suggests, a diagnostic package (program, or utility) is a tool for determining how well a PC is functioning, and for helping to pinpoint the cause of problems exhibited. There are many diagnostic packages available for personal computers, each with its own area of emphasis, and each with its own array of advantages and limitations.

Our attention here will focus on the classes of diagnostic features intended for use by the average or advanced PC user, rather than those designed for highly trained computer technicians. Being menu-driven and offering context sensitive help, the user-oriented packages are in general simple to operate, and the results obtained are fairly easy to understand.

Such packages are of value for at least four reasons. First, they provide an easy way to obtain a rather comprehensive record or baseline on a PC before problems occur. As noted in Chapter 9, an important part of troubleshooting is to have on hand a record for each machine which provides information about the make and model of the PC. This includes the particular BIOS it is using (along with information on other key chips and components), the optional equipment that is installed (adapters and peripherals), and the configuration settings (whether by DIP switches and jumpers, or as resident in an internally stored table).

It also should include some measure of how well the system performs when it has no apparent problems. In a similar vein, the package affords a quick and easy way to find out about a machine with which you may not be familiar.

Second, for anyone just beginning in the field of PC maintenance, a good diagnostic package is an excellent instructional device. Assuming that the package is accompanied by a reasonably well written manual, much can be learned from seeing the software in operation and from the definitions and explanations in the manual.

Third, the package may be useful in actually diagnosing a problem. Of course, the machine must be healthy enough to allow the diagnostic package to be run, and there are many things outside of the package itself that the user must know about PCs in order to get the most out of the diagnostic tool, let alone to be able to correct the problem.

Finally, having the baseline record before problems occur and, if possible, being able to run the diagnostic package, the user is in a much better position to describe the problem or symptoms to a qualified technician if the difficulty cannot be easily resolved by the on-site diagnostician.

In general, the diagnostics packages available are organized into three major sections:

- Providing information about the system (hardware and software configuration)
- Providing an array of specific tests (memory, controllers, disk drives, etc.)
- Providing a set of measures of system performance called benchmarks. The benchmarks indicate how the speed, capacity and other parameters of the system compare to some arbitrarily selected standard machine, such as an IBM PC or XT.

A well-designed diagnostic package, furthermore, will permit the user to tailor the tests to his or her specific needs, and to save a record of the results on paper (printout), disk or both – in addition, of course, to seeing the results on the screen. A further particularly useful feature that may be provided by the diagnostic package is the ability to create a graphic layout depicting the actual chips and their locations on the motherboard. Once this is done, a failed chip can be easily identified.

Collectively, the features listed above will provide an excellent profile of the system and provide a frame of reference against which the magnitude of a problem and the success of its remedy may be assessed. In the remainder of this chapter, we will cover some of the more common elements of a basic diagnostic package. In so doing we will be revisiting many of the terms and functions of a PC touched upon in earlier chapters. If, for example, you have read the chapter on storing information on magnetic media, the notion of Write Pre-compensation as covered in the present chapter will be immediately understood.

Similarly, we have already addressed the subjects of BIOS, CPU, RAM, ROM, math co-processor, memory mapping, device drivers, disk platters, cylinders/tracks, read/write heads, and other topics which appear when a diagnostic package is run. So, the present chapter should carry your understanding a step further, while at the same time adding some new terms and functions to your array of troubleshooting skills.

Presented in Figure 11-3 is a listing of the diagnostic topics that will be addressed. Our intent is not to teach the use of a particular diagnostic package. Rather, it is to provide a preliminary tutorial that will make it easier to read, understand, and use any diagnostic package of the type covered here.

Fig. 11-3. Listing of Diagnostic Topics Covered

System Information	Diagnostic Tests	Benchmarks
System Configuration	Memory	Main system
DOS version	Program buffers	CPU speed
EMS version	Base memory	Math speed
BIOS manufacturer	Extended memory	Video speed
and release date	Expanded memory	BIOS video
Processor & bus type	High address lines	Direct video
Math co-processor	Hard disk	Hard disk
Memory Map	Controller card	Transfer speed
Base memory	Read tests	Seek time (Access time)
Extended memory	Floppy disk/drive	Average seek time
Expanded memory	Drive	Track-to-track seek time
EMS page frame	Floppy disks	
Video sub-system	System Board	
Video adapter	CPU	
EGA switch settings	NPU	
Video address and	DMA controller	
video RAM size	Interrupt controller	
Fixed disks	Real-time Clock	
Floppy disks	Current time	
Clock-calendar	Real time clock vs DOS clock	
Parallel ports	Compare elapsed time	
Serial ports	Serial Port	
Joysticks	Parallel Port	
Mouse	Printers	
	Video	
	Input devices	

System Configuration

The initial phase of the diagnostic package should examine the system and present a variety of important configuration information. Figure 11-4 was obtained by running the diagnostic package, "Checkit," (a product of TouchStone Software Corporation) on the machine being used to write this book. The data was saved directly to disk and then retrieved into the present document. The entire procedure as just described took about a minute. Here we have an excellent snapshot of the machine for future reference in troubleshooting, and as a possible inventory record. At a glance, we can see that the machine is a PS/2 (80386 MicroChannel) system. The machine happens to be an IBM PS/2 Model 70. As presently configured, the machine has no math coprocessor.

The ROM BIOS is IBM, with a release date of February 2, 1989. The video is a VGA/EGA class monitor, and the switch settings on the adapter board are 0110, meaning: open; closed; closed; open. The machine has a 121Mb hard drive; a 3 1/2 inch, 1.44Mb floppy drive; 640Kb of base (user) memory installed; and 3328Kb of RAM in the extended memory area. It shows also that no Expanded Memory Specification (EMS) driver is installed, and thus it cannot, at present, use any expanded memory that may actually be installed.

We see further that the system has a CMOS clock-calendar, and we know that this class of machine also stores its configuration data in a CMOS table. This model of PC comes with a reference diskette that will automatically reconfigure the system (restore the CMOS configuration settings) should the CMOS table be lost due to battery failure, or other mishap. In fact, a power surge did blow away the CMOS table while this document was being written. When the system was re-booted, error codes 162 and 163 appeared on the screen, indicating that the CMOS table and clock were "not OK" – the machine would not run. In just a matter of minutes, the autoconfigure routine on the reference diskette had things back in working order.

But what if the machine were some other model that did not have automatic reconfiguration by a software program? All of the configuration data would have to be available and entered from the keyboard, using the PC's setup program. Once again, a strong case can be made for using the diagnostic package to obtain and record vital information about the system – before problems occur!

Fig. 11-4. Configuration Table Produced by Checkit

DOS Version:	5.00
ROM BIOS:	IBM BIOS Date: 02/20/89
Processor Type:	80386 Micro Channel
Math Coprocessor:	Not Present
Base Memory:	640K (Available: 572K)
Extended Memory:	3328K (Available: 3323K)
EXPANDed Memory:	No EMS driver installed
Video Adapter:	VGA (EGA Switches: 0110)
Video Address:	A000h (Video RAM Size: 256/512K)
Hard Drive(s):	Drive 0 (C:) = 121M
Floppy Drive(s):	A:1.44M (3.5")
Clock/Calendar:	CMOS Clock
Parallel Port(s):	LPT1=3BCh
Serial Port(s):	COM1=3F8h
Mouse:	None; Joystick(s): No Game Port

With the basic configuration data in hand, we may now turn our attention to some of the detailed data, tests and benchmark measures in a typical diagnostics package, beginning with a look at the amount and type of memory on board.

Memory Map

The memory map shows how the first 1Mb of address space is used. (See Chapter 3 for a discussion of memory mapping.) The first 640K is user memory, and the remaining 348K of the first 1Mb (1024K) comprises the reserved area, and that RAM used by the video display lies within the reserved area. Base memory is the amount of RAM actually installed. If base memory is less than or equal to 640K it occupies the user memory area. Any RAM greater than 640K is mapped (addressed) above the 1 Mb level. The map indicates the addresses of the items listed below ("Interrupt Vectors" through "System ROM").

Interrupt Vectors (Software)

Interrupts are of two kinds, software and hardware. Software interrupts allow programs to access DOS and the BIOS. They are different from the Interrupt Request Channels (IRQs) used by the hardware devices on the system, as described further on.

Resident Programs

This feature of the diagnostic shows the programs currently loaded (resident) in the user memory area. This would include DOS, any TSR (terminate and stay resident) programs, and the memory space occupied by the diagnostic program itself. The amount of unused (available) user RAM would also be indicated.

Extended BIOS area

This is memory (1K) assigned at the upper end of the user memory address range. It is used by OS/2 (if you are running OS/2, a multi-tasking operating system, rather than DOS).

Video RAM

The video adapter board on your system requires memory space to store text and graphics for display on your monitor. The amount of RAM used varies from one type of display board to another. A typical amount of RAM here would be 128K.

EMS Page Frame

The EMS (Expanded Memory Specification) page frame is the window, usually 64K in size, through which data may be sent to and retrieved from the expanded memory area.

Adapter ROM

Video adapter boards and certain hard disks contain ROM (read-only memory). The diagnostic program should indicate the presence of the adapter ROM and provide some identifying information about it.

System ROM

Every PC has a basic input-output system (BIOS), which is necessary for booting up the system and for communicating with system hardware and peripherals. The BIOS is stored on a ROM chip, and is referred to also as the "system ROM." The diagnostic program should display the name of the BIOS manufacturer and the release date of the ROM on your machine.

Interrupts (Hardware Interrupt Request Channels, IRQ's)

When a "hardware event" occurs, such as pressing a key on the keyboard, the device uses its assigned interrupt request channel (IRQ) in order to access its driver. A driver is a software program that allows hardware (e.g., keyboard, mouse, LAN cards, tape drives) to communicate with the system. These drivers are TSR (terminate and stay resident) programs. They must be loaded in order for the diagnostic program to detect them.

Usually the driver programs (device drivers) are loaded from within the CONFIG.SYS file, although some may be loaded from the AUTOEXEC.BAT file. Note that when diagnosing a system, especially where memory conflicts are the suspected problem, it may be necessary first to boot the system from a "virgin" floppy disk (containing no CONFIG.SYS or AUTOEXEC.BAT file) and then introduce (load) the device driver files one at a time in order to isolate the source of the memory conflict. The diagnostic program should list all of the device drivers present, and no more than one device should be assigned to a given IRQ channel. Typical PC interrupt (IRQ) assignments are shown in Figure 11-5. (Variations are possible.)

Direct Memory Access (DMA) Channels

A direct memory access channel (DMA) is a path whereby a device (such as a disk drive) can write information directly into memory without having to go through the

CPU. The advantage of bypassing the CPU is speed of data transfer. The transfer process is handled by a DMA 8237 controller chip. Each such chip controls four DMA channels.

Fig. 11-5. Typical Interrupt (IRQ) Assignments

Interrupt Line	Device
0	System Timer
1	Keyboard
2	DMA Controllers (Cascade)
3	Available
4	Com1
5	Available
6	Floppy Disk
7	LPT1
8	Clock/Calendar
9	VGA (Active)
10	Available
11	Available
13	Reserved for Math Coprocessor
14	Hard disk
15	Available

The IBM PC and XT (machines having 8 bit bus slots) each has one DMA controller chip; IBM AT class machines (machines having 16 bit bus slots) have two DMA controller chips, for a total of 8 channels. When two DMA chips are present, they are connected in tandem (one behind the other). This arrangement is referred to as a "cascade" configuration. Thus, when you run a diagnostic program, you may see the word cascade on the report screen. Typical DMA assignments for PCs are shown in Figure 11-6.

CMOS Table

CMOS memory is powered by a battery and thus retains its information when the PC power is turned off. A major application of CMOS technology is on the AT and 80386 class machines, where it stores important configuration in the "CMOS table." IBM PCs and XTs used DIP switches to set up the configuration – i.e., to inform the system about the equipment installed on it. Later machines (AT and '386 machines) use software and a setup menu to input the configuration information which is then stored in the CMOS table (chip).

Fig. 11-6. Typical DMA Assignments

DMA Channel	Used for:
0	Dynamic refresh of RAM
1	Hard disk controller
2	Floppy disk controller
3	Not used
4-7	On AT and PS/2 machines; usage varies.

The following items ("Current Date and Time" through "Hard Drives") may be found in the CMOS table, as typically displayed by a diagnostic program. Note, however, that most of the settings (values in the table) cannot be changed using the diagnostic routine. To input or change the settings, you must run the setup or configuration utility that comes with the PC.

Current Date and Time

This is the current date and time of the system "real time" clock. This clock is run by a battery and thus remains current so long as the battery is functioning. When DOS boots up, it uses this date and time.

Floppy Drive A: and B:

The diagnostic displays the type and capacity of floppy drives installed.

Base Memory Size

The diagnostic displays the actual amount of user RAM installed (configured) on the machine. Note that the physical installation of RAM chips alone is not sufficient. The system must be configured (told) as to how much RAM is installed.

Extended Memory Size

Extended memory is the area of RAM lying above 1 Mb. If RAM chips are installed to provide memory above 1 Mb but they are not configured, the additional memory (which can be as much as 15 Mb) most likely is not being used.

Primary Display

Here, the type of video adapter that is configured as the primary display (EGA, VGA, etc.) is indicated. If the display were not configured correctly, it would not be operating.

Hard Drives

The type of hard drive(s) installed are indicated, along with the drive "parameters." The parameters are the drive descriptors and specifications the system must know about in order for the drive to function. The types of drives supported by your system are contained in the BIOS ROM. When the setup utility is run, a table displaying the types of drives supported will be displayed. If you have one of the controller/drive combinations shown in the table, all that is necessary is to specify the drive type number. Otherwise, it will be necessary to input the specifications for each column, or to select a type that most closely matches the controller/drive you have.

Figure 11-7 illustrates one row of a drive parameter table. Column order and abbreviations may vary. Each of the column heading abbreviations is explained below.

Fig. 11-7. Sample Row of a Typical Drive Parameter Table

Type/No.	Cyl	Hd	WPC	Ctrl	LZ	S/T	Cap/Meg
13	306	8	128	00h	319	17	20.32

Type/No – The number assigned to designate a drive having the parameters shown across the remainder of the row in the table. In this example, the type/number is "13."

Cyl – The number of cylinders on the drive. In this case there are 306 cylinders.

Hd – The number of read/write heads on the drive (or the total number of tracks per cylinder). In this case, there are a total of 8 heads.

WPC – Write PreCompensation starting cylinder. Here, it is cylinder 128. See note below for an explanation of WPC.

Ctrl – Control byte (00h = not used)

LZ – Landing Zone cylinder for parking the heads (e.g., cylinder 319)

S/T – Sectors per Track. Here there are 17 sectors per track.

Cap/Meg – Disk capacity in megabytes (Mb). Here the capacity is 20.32 Mb.

The diagnostic program should display the parameters for the hard drive as they are currently configured on the system. A "Type 0" indicates that no hard drive is installed, or that a special type of controller may be in use (e.g., ESDI; SCSI). A "Type 47" drive is one for which you must enter the parameters, as the drive table does not include an entry for the controller/drive on the system.

__Note__: The tracks (or cylinders) closer to the center of a disk are smaller in circumference than those near the outer edge. Thus, the data (magnetic charges) must be placed closer together on the inner tracks. Recall from Chapter 6 that magnets can attract and repel each other, thus the magnetic charges placed close together can cause each other to shift or "migrate" apart. To compensate for this shift ahead of time when the data is written to the disk (i.e., to "write pre-compensate"), the magnetic charges are placed closer together such that the migration is taken into account, and so the data will be able to be read. Not all drives require write precompensation to be specified in the setup table, however.

Device Drivers

Device drivers are software programs that provide an interface between the computer and the hardware devices attached to it. In a DOS-based system, devices are indicated by the use of a colon (:). That is why you see the hard drive, e.g., the "C" drive, written as C:, the A drive as A:, the Line PrinTer as LPT1:, the communications port as COM1:, and so on.

There are two classes of device drivers (i.e., "devices") – those that handle one character at a time, and those that handle blocks of data (or characters) at a time. Thus, we have "character devices" and "block devices." An example of a character device is the keyboard, because it allows you to type (input) only one character at a time into the system. An example of a block device is a disk drive, because it transfers data in blocks (chunks; groups of characters) at a time. A diagnostic program typically will display a list of each class of device (character and block), along with the hexadecimal address (segment and offset) of each device and other descriptive data.

Diagnostic Tests

Memory Tests

Program buffers – checks operation of memory storage space (buffers) used by programs.

Base memory – Checks operation of memory installed in the user memory area (whatever is actually installed, up to a maximum of 640Kb of user RAM).

Extended memory – Checks RAM in the extended memory area.

Expanded memory – Checks RAM in the expanded memory area.

High address lines – Tests for address-related problems that may be caused by the presence of an incorrect chip, or by broken or bent pins on a chip.

Fixed Disk (Hard Disk) Tests

"Read" tests do not write any data to the hard drive; they only read what is already written. However, they do also test the readability of any disk space that is not currently occupied by data or documents. The tests should report any places on the disk already marked by the low level format or by the high level (DOS) formats as unusable.

A report of a "soft error" indicates that a track may be starting to fail. An error associated with "free space," i.e., space that could potentially be used by DOS to write your data to the disk, is cause for concern. If such an error occurs, the entire disk should be backed up and reformatted using the DOS Format command. The reformat should cause the bad area on the disk to be marked as unusable.

The hard disk test should test the disk controller board first. If that passes, it should proceed to at least the three basic read tests (linear, butterfly, and random) as noted below.

Fixed Disk (Hard Disk) Controller Card

Tests operation of the hard disk controller card. An improperly functioning controller card can cause read/write errors and/or contaminate data on the hard disk.

Read Tests

"Read tests" determine how well data on the disk can be read and retrieved. Each cylinder and read/write head on the drive is tested. There are three basic kinds of tests, as explained below.

Linear Read. A linear read test examines each cylinder in order, beginning with the first cylinder (which is cylinder 0, at the outer edge of the disk) and working one cylinder at a time toward the center (inner cylinders) of the disk.

Butterfly Read. In contrast to the linear read as noted above, the butterfly test starts with the outer cylinders and then jumps to the inner cylinders, working back and forth progressively between the two. This is a stringent test of the drive's ability to seek out specific tracks, and to reveal problems associated with head movement and alignment of the cylinders.

Random Read. Recall from Chapter 6 that data is usually not stored on contiguous cylinders on the disk. As data and documents are created and erased during normal use of the PC, DOS writes any new data or documents in whatever space happens to be the next available on the disk at the moment. Therefore, the disk heads may be required to move from one place on the disk to another in a relatively non-systematic sequence. The random read test is designed to represent normal disk activity more realistically that either the linear read or the butterfly read tests.

Floppy Drive and Diskette Test

The floppy drive tests will test both the drive itself, and will further allow specific diskettes to be tested. In addition to a read test, the drive's ability to write data is also checked. Since the write test causes information to be placed onto a disk, a disk should be used only if it does not contain data that you wish to keep safe. The disk used in the write test (which is testing the drive) should be formatted to its full capacity using the DOS FORMAT command, before beginning the test.

Test the Drive

The drive is checked using a random read test (as noted earlier under "hard drive" tests), and a random write test. Although the order of the read and write tests may be random, the ability of the drive to use the entire disk should be tested. That is to say, a "random" test does not imply that only a random sample of the disk area has been tested. Note that it is possible for errors to be reported that are due not to the drive but to the disk used for the test. If problems are suspected with the drive itself, re-run the tests using another disk to verify that the problem is indeed with the drive.

296

Test Specific Floppy Disks

If you are testing a specific diskette, it should be formatted to whatever capacity will be used on it. Drives of differing capacity (for example, 1.44Mb vs 750Kb, 3 1/2 inch drives) also differ in the width and strength of the magnetic track created on the disk surface. This can give rise to problems, for example, when attempting to move disks from drives of one capacity to another. As a general practice, therefore, diskettes should be formatted to the full capacity provided by the type of drive in which they will be used.

System Board (Motherboard) Tests

System board tests include those for the CPU (Central Processing Unit, also called the system "microprocessor"); the NPU (Numeric Processing Unit, also called the numeric co-processor or math co-processor); the Direct Memory Access (DMA) controller); and the interrupt controller. All of these are chips on the system board.

Figure 11-8 provides useful reference information on CPUs and NPUs, and the types of IBM machines (and clones) in which they are found. Note that for each CPU there is a corresponding type of NPU to be used. CPUs and NPUs must be correctly paired or the system will not operate properly.

Fig. 11-8. CPUs, NPUs, Data Bus Size, and Examples of Computers Using Them

CPU	NPU	Data bus (bits)	Example of models in which CPU is found
8086	8087	16	PS/2 Model 30
8088	8087	8	IBM PC, XT, Portable
80186	8087	16	
80188	8087	8	
80286	80287	16	IBM AT; PS/2 Model 50, 60
80386DX	80387 or 80287	32	PS/2 Model 70, 80
80386SX	80387SX	16	
80486SX	80387 built-in	32	
80486DX	80387 built-in	32	PS/2 Model 95

CPU chips are designed to operate at or below a specified speed (clock rate). The chips usually have the clock rate marked on them. The marking typically shows the manufacturer's name, the chip type (8086, 80286, etc.), with the number followed by a speed code number (suffix). For example, "Intel 80386-16" indicates a chip manufactured by the Intel Corporation. It is an 80386 chip designed to operate at a maximum speed of 16 megahertz (MHz). Figure 11-9 shows the maximum clock speed (MHz) for various CPUs.

Fig. 11-9. Maximum Clock Rates for Various CPUs

CPU	Clock Rate Suffix	Maximum Speed (MHz)
8086	(none)	5
8088	-3	6
	-2	8
	-1	10
80286 and	-6	6
80287	-8	8
	-10	10
	-12	12
	-16	16
	-20	20
80386 and	-16	16
80386SX	-20	20
	-25	25
80386DX	-20	20
	-25	25
	-33	33
	-40	40
80486SX	-20	20
	-25	25
80486DX	-25	25
	-33	33
	-50	50

In running a diagnostics package, two known problems ("bugs" in the chips) may be encountered for early releases of specified CPU chips as follows:

- 8086 and 8088 – Interrupts: chip allows interrupts to occur at the wrong time; machine can hang up without warning. If found, a defective chip should be replaced.
- 80386 – 32-bit multiply routine: produces inconsistent results; machine may hang up intermittently; probably will not appear when running 16-bit operating systems such as DOS and OS/2; may occur when running 32-bit code, such as found in Windows/386, Unix/386, and memory management written for 32-bit systems. A defective CPU should be replaced.

CPU (Central Processing Unit)

The test examines key functions of the CPU (also called the system "microprocessor"). Operations tested should include CPU logic, math and general functions; tests for known bugs (as noted above); and real vs. protected mode operation (explained below). Speed of the CPU is measured under the topic of "Benchmarks," addressed later in this chapter.

A machine with an 80286 or 80386 CPU is capable of running as a multi-tasking environment. This means that more that one application can be loaded and used at the same time. In this environment one could, for example, load and work with Word Perfect, and at the same time have Lotus 1-2-3 loaded – and be able to switch back and forth between the two. To do so, however, it is necessary to have the PC running under a multi-tasking operating system, such as Unix or OS/2. DOS is not a multi-tasking operating system. It runs only one application at a time. Therefore, the 80286 and 80386 machines have been designed to operate in two different modes. When they run under DOS, they do not use the full capability of the CPU. Instead, they emulate the 8088 CPU. This is the so-called "real" mode. The name "real" mode appears to have no logical semantical basis.

In the "real" mode, (running under DOS and 8088 emulation), expanded memory cannot be directly accessed, as it might be using the full capability of the '286 or '386 CPU. Furthermore, DOS "assumes" that only one application is running and that DOS itself has access to whatever user memory it needs at any time. In this sense, the use of such memory is "unprotected" – it is assumed to be available for whatever demands that DOS may make upon it. In essence, there is no second or third application being run that DOS needs to worry about.

On the other hand, when a multi-tasking operating system such as Unix or OS/2 is being used, the PC does not emulate the 8088 CPU mode of operation. All of the user memory (up to DOS's usual 640Kb of user memory) as well as extended memory can be used. This requires more oversight of memory usage by the operating system – it must be sure that no conflicts occur in the demand for or use of memory. So, it "protects" the memory being used for various purposes at any given moment. Thus, the operation is referred to as the "protected mode."

A PC may operate without problems in its "real" (i.e., DOS and 8088 emulation) mode, but there could be a mishap awaiting use of the protected mode, for example, when you add extended memory and software that can use it. Therefore, the diagnostics package should check PC operation in both modes.

NPU (Numeric Processing Unit; Math Co-Processor)

Tests various floating point calculations and comparisons. Speed of the NPU is measured under subject of "Benchmarks," covered later in this chapter.

DMA Controller

IBM PC/XT machines have one, 8237 DMA controller chip. AT and PS/2 machines have two DMA chips, arranged in tandem (cascaded). The DMA controller test should check communications between the CPU and the controller, controller channels, registers and refresh rate.

Interrupt Controller

IBM PC/XT machines have a single, 8259 interrupt controller chip; AT and PS/2 machines have two interrupt controller chips. The job of the interrupt controller is to interrupt the current activity of the CPU so that a hardware event, such as a key being pressed on the keyboard, can be processed. An example would be when the ESC key is pressed to halt a program that is running.

Real-time Clock Test

The real-time clock is the battery operated circuit that keeps track of the calendar date and time of day when the PC is powered off. The "DOS clock" is the clock circuit that runs under DOS when the PC is powered on. The DOS clock takes its setting (date and time) from the real-time clock when the system boots up. If the PC does not have a battery operated clock, the DOS date and time must be set each time the PC is powered up.

Current Time. Shows the current time, read from the real-time clock

300

Compare Real Time Clock vs DOS Clock. Compares time on real time clock with that on the DOS clock. The real time-clock and the DOS clock should agree within a few seconds deviation.

Test Teal-time Clock Alarm. The real time CMOS clocks of AT and PS/2 type machines (if installed) have an alarm function. This is not an audible alarm – it does not produce a "beep." Rather, it is an interrupt function.

Compare Elapsed Time. To be reliable, any two clocks should keep time at the same rate. A difference between the elapsed time on the real-time (CMOS) clock and the DOS clock may indicate that the CMOS clock battery is beginning to fail.

Serial Port Test

The serial port is used mainly for interfacing with modems. The word "modem" is a contraction, based on the words "MOdulator/DEMmodulator." A modem allows data to be sent over telephone lines from one computer to another. A serial port transmits one bit of data at a time (i.e., a series of bits), in contrast to a parallel port which transmits one byte at a time (i.e., 8 bits in parallel). Serial ports are referred to as "COM" (communications) ports. Other names for the serial port are RS232, RS232C and EIA 232D. All are the same thing, and all are "asynchronous" ports. The asynchronous ports are usually found on a multi-function adapter board installed in one of the adapter slots on the system board. These ports use either a 25 pin connector (DB25) or a nine pin connector. In either case, only nine pins are used. Problems are often associated with the cables and connectors, either because the cable/connector assembly is wired incorrectly, or because of loose or damaged connections.

The RS232 is a bi-directional port. That is, it sends (outputs) and receives (inputs) information. The port can be tested by diagnostic software, which may or may not require the use of a "wrap plug." A wrap plug is an external loopback connector (i.e., a device or "plug" that connects to the port) such that information output by the port can be sent directly back into the input side of the port. The test then compares what is sent out with what comes back in. The output should match the input or there is a problem.

Asynchronous RS232 ports generally are based on four chips. The main chip will be either a UART 8250 chip (older machines) or a UART 16450 chip (late model machines). UART stands for "Universal Asynchronous Receiver/Transmitter." The other three chips consist of two 1488s and one 1489. These three chips are "driver" chips. The 8250 may be replaced with a 16450. Note that OS/2 requires a 16450 chip.

The diagnostic test should check the port input and output functions, and the chip registers. Use of a wrap plug is preferred, because it more faithfully represents actual conditions of port use. The test should also indicate the port's baud rate. "Baud" is a unit of measure of data transmission rate (named after J. M. E. Baudot). Baud is actually a number indicating the rate-of-change-per-second of the transmission signal. The baud rate is roughly equivalent to data transmission expressed in bits per second (bps). Technically, however, baud and bps are not the same thing. Common baud rates are 300, 1200, 2400 and 9600.

Parallel Port Test

Parallel ports are used mainly for (parallel) printers. Thus, they are commonly designated as "LPT" ports, standing for "Line PrinTer." The main advantage of using parallel ports with printers is in the speed of data transmission. A limitation of parallel transmission, however, is the length of cable between the PC and the printer. Since parallel transmissions are vulnerable to electronic interference, the cable should be kept relatively short (i.e., under 15 feet). Usually, the transmission is unidirectional (one-way), with output from the PC being sent to the printer. On the PS/2, however, the parallel port is bi-directional.

The diagnostic test should check the output of the parallel port and the port registers. Again, as with testing the serial port, a wrap plug is to be preferred for the test.

Printers

Most printers have a built-in, self-test (explained in the manual that accompanies the printer). If a printer problem is suspected, this test should be run first, with the printer disconnected from the PC. Printer problems, however, may be due to faulty or incorrect cable connections, or to incorrect configuration of the PC. That is, the setup, drivers, font cartridges, etc., selected on the PC do not match the printer connected to the PC.

Most diagnostic software packages will provide a "generic" test, along with tests for several of the more popular impact printers (daisy wheel and dot matrix) and photocopy type ("laser") printers. The printer tests are straightforward and so they will not be detailed here. Additional discussion on printer operations is provided in Chapter 16.

Video

Most of the video tests are concerned with the quality of the image displayed, both text and graphics; and for color monitors, the purity of the colors. Typically the test displays a sample or pattern on the monitor, and the user must judge whether it is acceptable. A further important test is for the video RAM. This is the RAM provided on the video adapter board itself, and should not be confused with the video (system) memory allocated within the reserved area of the system memory map.

Input Devices

The diagnostic package will most likely include tests for the most commonly encountered input devices: keyboard, mouse and joystick. Note that each of these devices typically uses a serial port to interface with the PC.

Keyboard Test

Keyboard tests will check the functioning of each key, and whether the keyboard lights are functioning properly. Three main types of keyboard are the PC/XT keyboard, with 83 keys; the (older) AT keyboard, with 84 keys; and the enhanced keyboard, with 101 keys.

Mouse Test

The mouse test should include proper functioning of the mouse buttons and control of the mouse cursor across all areas of the display monitor.

Joystick Test

The joystick test should include proper functioning of the joystick button(s); control of the cursor across all areas of the display monitor; and centering of the cursor on the display relative to the physically centered position of the joystick in its mounting.

Benchmarks (Performance Measurement)

The benchmark portion of a diagnostic package measures and reports how well (i.e., how fast) certain key elements of the system are performing – the CPU, NPU (if installed), video, and the hard drive. This information is useful to verify the functioning of a new system, to provide a performance record of the machine for later reference, and to determine the source of certain problems if they exist. Results are expressed in absolute units (e.g., characters per second, Kbytes per second), in relative units (e.g., as "so many times faster than an IBM PC/XT"), or both.

CPU Processing Speed

CPU processing speed is measured in "dhrystones." Performance is expressed relative to that of a specified computer, such as an IBM PC/XT. Actual processing speed varies according to the specific task; total work productivity speed depends on all of the other components used (e.g., video; disk drives).

Arithmetic (Math) Processing Speed

Arithmetic processing speed is measured in "whetstones." The diagnostic should report the CPU math processing speed; and, if an NPU (math coprocessor) is present, its speed and type should also be reported.

BIOS Video Speed

BIOS video speed is measured in characters per second (CPS). This is the speed at which information is displayed on the monitor, where the information is sent from the CPU to the monitor via the BIOS. This method is used mainly on older systems and certain types of terminal emulations.

Direct Video Speed

Direct video speed is measured in characters per second (CPS). Display data is not run through the BIOS; it goes directly from the CPU to the video. This is much faster that the BIOS video mode; most current applications employ direct video.

Hard Disk Performance Measures

Transfer Speed

The single most important question is how fast the drive/controller combination can transfer data from the disk to the system board. This is referred to as the hard disk transfer speed (or, transfer rate), expressed in Kbytes per second. To transfer data, the drive must access ("seek") the needed cylinders and position the heads to access the data. Therefore, a few other measures of disk performance and/or configuration are obtained, as noted below.

Access (Seek) Time

Access time is the time needed (on average) to access a given cylinder. This is usually measured as the time needed for the heads to move across one third of the cylinders, expressed in milliseconds.

304

Track-to-Track Seek Time

This is the time needed (on average) for the heads to move from one cylinder to an adjacent cylinder, expressed in milliseconds.

Interleave Factor

DOS reads data from a disk according to the numbered sequence of the sectors on the disk. Since many disks spin too fast for adjacent sectors to be read one after the other, the sectors are numbered out of order. For example, sector 1 and sector 2 may be separated by two, three or more other numbered sectors. This gives the system time to read and transfer data from sector number 1 before sector number 2 spins into position under the read head.

The number of actual sectors separating any two consecutively numbered sectors plus 1 is called the "interleave factor." For example, if there were three sectors between sector number one and sector number two, the interleave factor would be 3 + 1, or 4. There is an optimal interleave factor for each drive/controller and system configuration. Any other interleave factor will degrade the drive/controller data transfer rate. (See Chapter 6 for a discussion of interleaving.)

Disk Cache Software

Certain applications may involve reading the same information repeatedly from a disk. If such information is to be used frequently, a faster method is to store (cache) it in RAM on the system (for example by using RAM in the extended memory area). Information thus stored is called "disk cache" information. XT and AT systems (and above) can support software that establishes and manages disk caches, and thus speed up operations by reducing the number of times the disk needs to be accessed during a work session. The diagnostic software should indicate the presence and location of any existing disk cache.


12
INSTALLING ADAPTER BOARDS

CHAPTER 12: INSTALLING ADAPTER BOARDS

Purpose and Nature of Adapter Boards

IBM personal computers (and their clones) are designed to use a variety of optional devices. In one sense, a device may be "optional" in that it does not have to be installed unless the user wants it (for example, a mouse or a modem). In another sense, a device must be present in order to operate the computer (for example, a keyboard and a video display), in which case the term "optional" refers to the make and model of the device selected by the user, rather than whether or not to have one at all.

All such devices need a way to communicate with the computer: a way must be established for control signals and data to flow between the device and the computer. A video display, for example, may be plugged into a wall outlet and it will light up, but it will otherwise be useless without a line of communication to the computer. Moreover, the physical design of a peripheral device such as a video display requires that special electronic circuitry be present, somewhere, to "interface" the device and the computer, or in other words, to "adapt" the device to the computer. The circuitry just mentioned typically resides on a printed circuit board, called an "adapter board," "adapter card," "add-on board," "option" board, or "controller." A board that provides input/output channels for communications is called a "port," but it is nonetheless an adapter. All are interface devices: the board plugs into a slot on the computer system board (motherboard), and a cable runs from the adapter board to the device it supports (e.g., the display monitor). Thus, the adapter board sits "between" the computer and the video display, keyboard, mouse, printer or other such "optional" device.

It is not always necessary for the interface circuitry to be constructed on a printed circuit board. For a variety of reasons, the same functions may be accomplished as well or better by putting the circuitry into a ROM chip, and mounting the chip to the system board. Recognize that much of the electronic circuitry involved here is to perform logical functions such as, "If condition X exists, then do Y; otherwise, do Z." These functions can be achieved by writing software code (logic instructions) into a ROM chip.

ROM chips retain their information when the system power is turned off, so their information and functions are preserved just as if they were rendered by hardware on a circuit board. For example, the adapter that sits between the keyboard and the computer is in the form of a ROM chip, rather than as an adapter board. Both the ROM and the adapter boards in this instance perform the same general functions, namely adapting (interfacing) the external device to the inner workings of the computer. In this chapter, our discussion will focus primarily on adapter boards.

Configuring the Adapter

Plugging an adapter into a computer and checking to see that the adapter and its peripheral device function properly is a relatively simple matter, and the subject will be addressed further on. But before an adapter is placed into the computer, several matters must first be attended to. Collectively, these steps are referred to as "configuring" the adapter. In essence, such configuration amounts to setting a few switches or jumpers on the adapter card. Why is this necessary?

The answer is that, on any given computer, there will usually be several such adapter boards installed – one for a video, another to provide serial and parallel communications ports, others for the floppy drive(s) and hard drive(s), and possibly adapter boards that provide additional memory. These devices, like new employees showing up on the first day of work, need to know "where to park;" "where to hang their coat;" "how to get attention (support) when they need it;" and "where to put their working materials." The devices and their communications channels, like the employees, are considered "resources" of the system.

Each employee may be capable of functioning quite well, but without assigning work locations, priorities and channels of communication, confusion and chaos would result. If the employees were to compete for the same parking space, channels of communication and so on, there would be "resource conflicts." Indeed, most adapter boards are able to operate perfectly when they arrive from the factory, but when they are plugged into the computer, they may not work because of the aforementioned resource conflicts. So, assuming that a board has no defects, the problem of board installation resolves mainly to that of configuring the adapter before it is placed into the computer (and then configuring the system, as will be discussed further on).

The functions accomplished in configuring a board are quite simple, but the names given to these functions can be somewhat intimidating. For the most part, at least as far as the average user is concerned, the functions themselves are not difficult to understand.

There are four elements to the configuration of an adapter board as shown below.

- Interrupt Request Channel (IRQ)
- Direct Memory Access (DMA) Channel
- Input/Output Address (I/O Address)
- ROM Address

IRQs and DMAs are of most relevance to our discussion; the remaining two items (I/O addresses and ROM addresses) will be covered briefly, just to round out the picture. The information at least will remove some of the questions that may arise when, for example, certain memory address information appears on a diagnostics report display.

Interrupt Request Channels (IRQ's)

Many activities are normally ongoing within a computer during a work session. Certain functions may be performed one after the other, and others are performed in parallel. The CPU is at the center of this activity, and the CPU is usually fairly busy – it doesn't just waiting to serve the keyboard (if and when a keystroke may be entered), or wait while a drive reads data from a disk and sends to the CPU.

When the keyboard, disk drives and other such devices need attention, they must interrupt the CPU from whatever it is doing. The process is not unlike calling the boss on his or her private line to request immediate attention. Now, if that private line is dedicated for receiving such "interrupt requests," it is, in effect, an "Interrupt Request Channel (IRQ)." In fact, the boss may have several such lines, i.e., a set of IRQ channels, and the computer has virtually the same arrangement between its peripheral devices and its CPU. And like the boss, who is but one person, the CPU can answer only one such line at a time. Therefore, the lines must be given an order of priority. Here, the solution is simple: the lines are numbered in order from 0 to 15, and the lower the number is, the higher the priority is – line 0 is first, line 1 is next, and line 15 is last in importance (priority) so far as interrupting the CPU is concerned.

Since most personal computers have a common complement of devices, to include the system timer, keyboard, serial communications port (COM port), floppy disk, clock/calendar, video monitor, and hard disk, and since at one time or another each of these devices needs to interrupt the CPU, certain IRQ channels are typically assigned to these devices by convention in the industry. A good way to see how this is done is to examine how an existing machine is set up, as we shall do momentarily. Also, if a new adapter is to be added to the system, it is necessary to see how the IRQ's for any existing boards are assigned. Otherwise, there is the risk of assigning an IRQ or a DMA to the new device, when that resource may already be in use by another device (and a resource conflict would therefore occur). The existing IRQ assignments may be found by running the setup or configuration software that comes with the computer. Another very convenient way is to run a diagnostic package, such as Checkit, and at the menu prompt, ask for ask a display of the "interrupts."

Shown in Figure 12-1 is the report produced bv Checkit 3.0 for the machine on which this book is being typed. The report was saved to disk and then called directly into this document.

Fig. 12-1. Interrupt Assignments

Checkit 3.0 Activity Log

INTERRUPT ASSIGNMENTS:		DEVICES WITH NO IRQ
IRQ 0	System Timer	– None –
IRQ 1	Keyboard	
IRQ 2	[Cascade]	
IRQ 3	Available	
IRQ 4	COM1	
IRQ 5	Available	
IRQ 6	Floppy Disk	
IRQ 7	LPT1	STANDARD DMA ASSIGNMENTS:
IRQ 8	Clock/Calendar	DMA 0
IRQ 9	VGA (Active)	DMA 1
IRQ 10	Available	DMA 2 Floppy Disk
IRQ 11	Available	DMA 3
IRQ 12	Available	DMA 4
IRQ 13	Reserved for NPU	DMA 5
IRQ 14	Hard Disk	DMA 6
IRQ 15	Available	DMA 7

As may be seen in Figure 12-1, there are 16 Interrupt Request Channels (IRQs) listed down the left side of the report, with the current IRQ assignments as indicated. We see that Channels 3, 5, 10, 11, 12, 13 and 15 are available for use. Note also that Channel 13 is reserved for a math co-processor (Numeric Processing Unit, NPU) should one later be installed. The existing assignments as shown are "standard." That is to say, Channel 0 (highest priority) is assigned to the system timer; the keyboard has Channel 1; and Channel 2 is given to the interrupt controller chips. In this case there happen to be two such chips, connected in tandem (cascaded).

The serial port (COM1) is assigned a fairly high priority (Channel 4), as would be reasonable considering that serial communications may need to interrupt whatever is happening in order to send or receive data. Lowest down on the totem pole is the hard disk (Channel 14). As we can see, there is nothing very complicated or mysterious about IRQ's. Basically, configuration tells each device which "private line" to use when it calls the CPU for assistance or attention.

Notice at the upper right of the figure the report also indicates whether there are any devices on board that do not need an IRQ. In this case, there are no such devices installed. By the way, Channel 7 is assigned to LPT1, which is a parallel port for a printer. The printer really doesn't need an IRQ, so if you are running out of channels, try using Channel 7.

Also shown on the report are the existing DMA (Direct Memory Access) channel assignments, so we shall consider them next.

Direct Memory Access (DMA) Channels

Certain devices, among them being the floppy disk controller (adapter), can write information (data) directly into memory, without having to go through the CPU. Writing the data directly to memory is much faster. For a device to do so, however, it needs to have a DMA assigned to it.

The DMA is analogous to an "electronic mailbox" – a "message" can be left "in memory" without having to "go through the receptionist or the secretary," i.e., the CPU. There are 8 DMA channels, numbered 0 through 7. The standard DMA assignment for the floppy drive controller is Channel 2, as shown in Figure 12-1.

But, just because a device is able to write data directly to memory, it is not absolutely necessary to use that capability by assigning the device a DMA channel. The device will still deliver its data to memory via the CPU. Normally, DMA Channel 0 is used for refreshing dynamic RAM (see Chapter 14), and DMA Channel 1 is assigned to the hard drive controller.

Input/Output (I/O) Addresses

Every device, such as a serial port, that needs to communicate with the CPU needs to know where it should look to find data to be sent, and where to deliver data that is received. The locations in computer memory for these functions are called "I/O addresses," and they are in the "high memory" area of the memory map (the area immediately above the 640K of conventional/user memory).

The I/O address concept is analogous to a mail room, where certain individuals (i.e., devices) are assigned a series of numbered boxes. The CPU can write information to one or more of the boxes for the device to send or use, and the device can write information to those boxes for the CPU to read. The boxes thus serve as an input/output medium. Each box has a number, i.e., an input/output "address." And, each device may be assigned a set of boxes – a "range" of input output addresses.

As with the IRQ and DMA channels, there are standard or conventional assignments of I/O address ranges assigned to the various devices. The addresses are given in hexadecimal numbers, but they are nonetheless just numbers like the numbers on post office boxes. For example, the address range for LPT1 (parallel port) on an IBM PC is 378h - 37Fh. Expressed as decimal numbers, the address range would be 216-223 9inclusive). Thus, LPT1 here has been assigned a total of 8 addresses. For a detailed explanation of hexadecimal numbers, please refer to Chapter 8.

Figure 12-2 lists a few devices and their hexadecimal I/O address ranges for the IBM PC.

Fig. 12-2. Examples of Device I/O Hexadecimal Address Ranges

Device	Hex Address Range
LPT1	378-37F
LPT2	278-27F
COM1	3F8-3FF
COM2	2F8-2FF
Floppy Disk Controller	3F0-3F7

312

Once again, we may take advantage of the Checkit diagnostic package and see where some of the I/O addresses are for devices installed on the machine being used to type this book. (Actually, what is shown is the starting address of the range of memory used.) The information is obtained by selecting the "Configuration" information on the Checkit application menu. Figure 12-3 shows the resulting report. The items of interest are shown here in boldface type.

First, we can see that the starting address of the video range is at A000h. Recall from Chapter 8 that hexadecimal numbers may be identified by an "h" or "H" tacked onto the end of the number. The "h" is not part of the number itself. The decimal equivalent of A000h is 640, thus indicating that the starting address here begins immediately above the 640K area of user (conventional) memory. Recall from the discussion of memory mapping in Chapter 3, that the user memory address range begins counting from zero rather than from 1. Therefore, the last (upper) address in the user memory range is 640K minus 1. This explains why the bottom of the reserved memory range begins at 640K rather than at 640K+1.

Also of interest, the report conveniently shows how the EGA switches on the video adapter card are set. As indicated, there are four switch settings: 0, 1, 1, 0, indicating OFF, ON, ON, OFF. We see further that the LPT1 (parallel port, printer) starting address is 3BCh, and the COM1 (serial port) starting address is 3F8h. You need not be concerned about setting these values. The matter is taken care of by the machine. They are noted here merely as a matter of general information.

All of the above memory locations are assigned by the system according to industry standards or conventions, and generally will not involve the user, unless two boards attempt to use the same address(es). Another exception would be where a sophisticated memory management application was to be used on the system, but that subject is beyond the scope of this book.

ROM Addresses

Certain adapter boards may contain information stored in ROM. The ROM data must be mapped to the computer's memory space (i.e., given an address in RAM) in order for it to be used. Typically, boards will have conventional memory ranges assigned to them, depending on the function the board serves (video, disk controller, etc.). Ordinarily, there is no problem with the addresses thus used. There is the possibility, however, that two different boards may be pre-set at the factory to use the same address range. If this happens, a resource conflict will occur. Certain boards will contain jumpers by which a technician can change the starting address, but this should not be attempted by the average PC user.

Fig. 12-3. System Configuration Information

Checklt 3.0 Activity Log

=== CONFIGURATION INFORMATION ===

DOS Version: 5.00	
ROM BIOS:	IBM BIOS Date: 02/20/89
Processor Type:	80386 Micro Channel
Math Coprocessor:	Not Present
Base Memory: 640K	Available: 572K
Extended Memory:	3328K Available: 3328K
EXPANDed Memory:	No EMS driver installed
Video Adapter: VGA	**EGA Switches: 0110**
Video Address: A000h	Video RAM Size: 256/512K
Hard Drive(s):	Drive 0 (C:) = 121M
Floppy Drive(s):	A:1.44M(3½")
Clock/Calendar:	CMOS Clock
Parallel Port(s):	**LPT1=3BCh**
Serial Port(s):	**COM1=3F8h**
Mouse: None	Joystick(s): No Game Port

Configuration Switches and Jumpers

Basically, configuring an adapter board amounts to setting a miniature switch and/or a jumper. Most such devices will require the assignment of an IRQ, and others may in addition make use of a DMA channel. Further, a serial communications adapter board (a "port" card) must be told whether it is to be COM1 or COM2, and a parallel board needs to be told whether it is to be LPT1 or LPT2. Note that COM and LPT port functions are commonly found on a single, multi-purpose adapter board, e.g., on a combination video and printer port card. The selection (configuration) is accomplished by setting the miniature switch positions on the board, and/or by putting a jumper across the appropriate pair of pins. Figure 12-4 shows a typical switches.

Fig. 12-4. A Typical Adapter Board Switch

The switch is actually a "block" of tiny switches. Usually there are 8 switches in a block. Some switches are "flat," where the setting is achieved by sliding a small tab forward or backward (to set ON or OFF). Others are "rocker" type switches, where the switch arm pivots at its center. When one end is pressed down, the other end comes up. The switches are constructed with their pins aligned in two rows, and are therefore referred to as Dual In-Line Pin (DIP) switches.

The directions for setting these switches are normally given as a string of ones and zeros, with each digit corresponding in sequence to one of the 8 switches on the block. For example, the setting might be prescribed as "10011000." These symbols should not be confused with binary numbers. They merely indicate the physical positions of switches as "on" or "off." A "1" indicates that the switch is to be set to "ON." A "0" indicates that the switch is to be set to "OFF." A switch that is "ON" is said to be "closed." A switch that is "OFF" is said to be "open." Sometimes the switches are marked to indicate which position is "1" (ON) and/or which position is "0" (OFF). Some switches may have no markings at all.

Accordingly, preserving and using the documentation provided with adapter cards and that of the computer itself is always a wise practice. Often, the only way to know which actions to take is from those documents. Once the board has ben configured, the next step is to install it in the computer.

Installing Adapter Boards Into Slots

A commonly asked question is whether a particular type of adapter board must be installed in a particular slot. in general, the answer is "no." Ordinarily, if the edge connector of a board fits the slot, the board will run in that slot. Be alert, however, to possible timing problems with the slot which is closest to the power supply on the IBM PC/XT. Avoid using the slot if at all possible, or use it for the IBM serial port card designed for that slot.

Apart from which slots may be used for which boards, an overriding consideration is whether the board can operate fast enough to keep up with the system; or, whether the speed of the system (i.e., the "slot speed") can be reduced to accommodate a slower ("speed sensitive") board. Boards that are speed sensitive include memory expansion boards, LAN (Local Area Network) cards and certain other types of communications adapters. Be sure to check on the clock speed(s) of the system in question and see that any adapter boards to be installed can operate at the speed(s) available.

Another question concerns the use of an 8-bit board in a 16-bit slot. Will the board work? Yes, it will. But, the practice should be avoided simply because it wastes the capability of the 16-bit slot. Also, there is the remote possibility of memory conflicts in the reserved memory area when both 16-bit and 8-bit boards are used on the same system, and when both types of board have ROM on them. Details are beyond the scope of the present discussion, but be alert to situations where an 8-bit board operates fine before a 16-bit board is installed, and then when the 16-bit board is installed, the 8-bit board no longer works. It is one thing for a newly installed board not to work, which may be due to any of several reasons. But, if a newly installed device appears to interfere with a device that was working well beforehand, be suspicious of a resource conflict (reserve memory, IRQ or DMA).

To physically install the board, turn off the PC's power switch, but leave the AC power cable connected, to provide grounding path for discharging static electricity. (See Chapter 14 for procedures and precautions when working inside the machine, and around any semiconductor chips, of which there are several on an adapter board.) Remove the cover, and touch a grounded surface such as the case of the PC power supply, to release any static buildup from your body.

Next, remove (and keep handy) the adapter retaining screw (and the slot cover, if present). Grasp the board at both ends and seat the edge connector into the slot, pressing down firmly. A slight rocking motion along the long axis of the board may help to ease the board into place. Then, replace the adapter retaining screw. (See Chapter 5 for details and illustrations on adapter installation.) Finally, connect the cable between the board and the device it supports. Always, follow the instructions provided with the adapter.

Configuring the System

When an adapter board is installed, it is necessary to "inform" the system about it. On IBM PC and XT machines this is done by setting miniature switches on the system board. On IBM AT and PS/2 machines the configuration is done using a software program. There are no switches to be set, with the exception of one switch on the AT for indicating that the monitor is either monochrome or color, and one jumper for the amount of memory installed.

Configuring PC and PC/XT Systems

The IBM PC has two blocks of 8 DIP switches (Block 1 and Block 2). The XT has only one block (Block 1). The position (on or off) to be set for each switch depends on what is installed on the machine. Figure 12-5 depicts the PC and XT switch blocks and indicates the purpose of each of the miniature switches (e.g., drives installed; monitor; amount of RAM). Switch Block 2 appears only on the PC, and it is used only for configuring the amount of RAM (base/user memory) installed.

To determine the correct position for each switch refer to the computers technical documentation and to the documentation provided with the adapter boards installed. The switch position may be set using a pointed object, such as a small screwdriver or toothpick.

Fig. 12-5. IBM PC and XT System Board DIP Switches

Configuring AT Systems

The AT has one switch to be set, to identify whether the monitor is a monochrome or color display. For a color monitor, the switch slide is positioned toward the front of the machine; for a monochrome monitor, the slide is positioned toward the rear of the machine. In addition, one jumper (J18) must be set when an additional bank of memory has been installed on the system board, and/or a 128Kb memory-expansion adapter has been installed. J18 is located at the right-hand corner of the system board toward the front of the machine. If the 128Kb expansion adaptor and/or a bank of memory has been added to the system board, the jumper should be across pins 1 and 2. Otherwise, it should be across pins 2 and 3.

To configure the AT after installing memory chips and/or adapter cards, and setting the J18 jumper if necessary (as described in the preceding paragraph), insert the Diagnostic Diskette that comes with the machine into the (A:) drive, and turn the power on. Assuming that the current configuration needs to be set (because something has been added, changed or removed from the system), the POST will report an error message. When this happens, press the F1 function key to continue. From there, follow the prompts on the screen and enter the data needed.

Configuring PS/2 Systems

On PS/2 systems, there are no switches to be set on the system board. Configuration is done using the "Configuration" option selected from the main menu presented by the Reference or Starter diskette that comes with the computer. Note, however, that PS/2 Model 25 and 30 need no configuration to be entered. The Reference/Starter diskette simply allows the user to confirm that the list of installed options is correct, set the date/time, and prepare the system for moving (park the heads).

Insert the Reference/Starter diskette into the (A:) drive and turn the system on. The POST will run, the system will boot up, and then the IBM logo will appear on the screen. Press the ENTER key to continue. The main menu of the Reference/Starter program will appear on the screen. Choose option number 3, "Set Configuration." If the configuration is not correct, the prompt, "Automatically reconfigure system?" will appear. Press "Y" to run the automatic reconfigure program. This should accomplish the reconfiguration.

Checking and Testing The Installation

Once the adapter itself has been configured and installed, and the system has been configured, the next step is to ensure that everything is working correctly. There are three main possibilities here:

1. Everything functions properly;

2. The newly installed device does not operate correctly;

3. The newly installed device operates correctly but a another device on the machine that was working properly before the new device was installed no longer functions. Each of these situations is discussed below.

Everything Functions Properly

So all is well, at least for the moment. But, electronic devices have a particular pattern of failure. If they are going to fail because of a manufacturing defect, they usually will do so early in their operating life, perhaps in the first hours or weeks of use. If they do not fail during that time, the chances are high that they will operate reliably for a long time. Barring abuse or a hostile environment, they may run for many years, whereupon their probability of failure increases as they essentially become "worn out."

An early failure is important to detect also because of the limited time that a warranty may be in effect for the device. Under normal use, a defective device might fail very soon after the warranty period ends. So, it is important to operate a newly installed device rigorously and continuously to determine whether it is one that will fail early in its life. This procedure is referred to as "burning in" the system.

A good way to "burn in" a device is to use a third party diagnostic package or test program. These packages allow you to select the function(s) to be tested, and to specify that a test be run repeatedly (perhaps overnight, or for a longer time).

If you happen to be an advanced user of a spreadsheet or word processing application you can, as a field expedient, create a test of your own by writing a program or a macro that exercises the functions to be tested. For example, you can create a large spreadsheet (e.g., in Lotus 1-2-3) and write a macro that repeatedly recalculates the sheet. The sheet should be large enough to use all of the memory available for it (base memory and, if installed, expanded memory as well).

Note that certain functions, such as the appearance of characters, graphics and colors on the display monitor, require a human observer to respond "O.K." or "Not O.K" during the various monitor tests provided in a diagnostics/test package. Here, there is no alternative to repeating the tests manually. It is not recommended that the operator sit in front of the machine for hours on end to do such tests. Rather, it would be important to run the tests several times when the machine is cold (e.g., when booted up after being powered off for an extended period, such as over a weekend), and again when the machine has reached its normal operating temperature.

The Newly Installed Device Does Not Operate Correctly

There are several main possibilities here: (1) incorrect system configuration; (2) incorrect adapter card configuration; (3) cables not connected properly; (4) incorrect slot speed (system runs too fast for the "speed sensitive" adapter board); (5) defective adapter board or a defect in the device it supports (e.g., monitor; disk drive; printer).

In general, the system configuration software for the AT and PS/2s should have detected a configuration error. If a problem still exists, recall the advice given in Chapter 10 on troubleshooting: retrace the installation steps; check the easiest things first; and isolate the components such that they can be tested with a minimum of interaction with or interference from other components. Alternatively, or in addition, running a diagnostics package such as Checkit may help to isolate the problem.

The Newly Installed Device Operates But Another Device Now Fails

If something on the machine was working properly until the new device was installed, there are two main possibilities: (1) when installing the new device, something on the other device (adapter card; cable) was changed or disturbed; or (2) a resource conflict exists.

Once again, retrace the installation steps, checking the obvious things first (is everything plugged in?), and isolate the components such that they can be tested with a minimum of interaction with or interference from other components. Then re-boot the machine and see whether any memory errors are reported.

A third-party diagnostic such as Checkit will help to identify resource conflicts. Be sure that there is not a conflict between IRQ assignments or DMA assignments. Resolution of any conflicts in ROM-to-RAM memory mapping will require expertise beyond that of the average PC user.

NOTES

13

INSTALLING DISK DRIVES

CHAPTER 13: INSTALLING DISK DRIVES

The installation of a disk drive is performed in two steps. The first step is to configure the drive. The second step is to physically install the drive in the computer. The physical removal and re-installation of disk drives is addressed in Chapters 4 and 5, respectively. In the present chapter, attention is focused mainly on drive configuration, which is essential to installing a new or replacement drive.

Drive Configuration and Installation in General

Configuration entails setting switches and jumpers on the drive itself. The procedure is not to be confused with configuring the system, which is the process of informing the system about the various options installed, and which is done after the drives have been installed.

Also, each drive (with certain exceptions) is equipped with a "terminating resistor network" (which may appear either in the form of a chip, or as a switch). The terminating resistor must either be removed or left in place, depending on where the drive is connected to the control and data cable coming from the drive controller.

The identifying markings of drive switches and jumpers (if any markings are present at all) vary a great deal from one drive to another – there is no one standard for these markings. It is essential, therefore, that the OEM (Original Equipment Manufacturer) be available for reference when installing a drive.

If, however, a drive has failed and is to be replaced by an identical drive, for example, by one taken from another machine, examine the switch and jumper settings on the failed drive carefully, and duplicate these settings on the replacement drive. Also note the presence or absence of a terminating resistor on the failed drive. If the failed drive has a terminating resistor, the replacement drive should also have a resistor (or the resistor switch must be set to open or closed, as appropriate.) This assumes that the replacement drive will be connected to the drive control and data cable exactly as the failed drive was.

The second step is to physically install the drive (mount it in the system, and connect the power, signal and ground connectors). PC/XT and AT systems use cables. Most PS/2 systems do not use cables. Note that where a drive is mounted to a metal chassis by metal screws, a separate, electrical ground connector may or may not be present or needed. The mounting screws may provide the contact for the system ground. However, drives mounted on non-conducting material (such as fiberglass rails) should have a ground connector wire. This ground connector is usually in the form of a tab or bayonet connector. Failure to connect the ground will likely result in problems with drive operation.

Further, it is important to ensure that the drive being mounted corresponds to the shape of the space in which it is to be mounted, including the alignment of the holes where mounting screws are to be inserted. The correspondence of drive shape and hole alignment is referred to in technical literature as the drive "form factor." Finally, the correct mounting hardware and screws must be used. Using the incorrect length and/or thread type of screws may damage the drive.

The procedure for installing floppy drives and hard drives is similar, but there are some important differences. Notably, floppy drives require more switch and jumper settings that do hard drives. Floppy drive installation and hard drive installation are addressed separately below. This is followed by a discussion of drive controllers and interfaces, and finally by a brief review of configuring the system after the drive has been installed.

Configuring and Installing Floppy Drives

Although drives may vary, there are usually four configuration options to be set. These consist of three jumpers, and a terminating resistor:

- Drive Selection (DS) jumper
- Terminating resistor
- Media Sensor (MS) jumper
- Diskette changeline/ready jumper

Drive Selection Jumper

On PC/XT and other ISA (Industry Standard Architecture) systems, two floppy drives may be connected to a single cable. The cable runs from the drive controller to one drive and then continues on to the next drive. This type of cable design is referred to as a "daisy chain." The purpose of the Drive Selection Jumper is to distinguish between the drives, designating one of them as (A:) and the other as (B:), and so that one or the other, but not both, will be activated by the disk controller at any given time. The last drive on the cable (i.e., the one farthest from the controller) must have a terminating resistor installed, and this is usually the (A:) drive. (The resistor is normally already installed on the drive when the drive is purchased.) The other drive, if present, must have its terminating resistor removed.

If the first drive (first in the order of lettering) is to be drive (A:), and the second drive is to be drive (B:), common sense would suggest that the jumper on the first drive (A:) be set to "1," and the jumper on the second drive (B:) be set to "2." Actually, the opposite settings are used. Thus, if the jumper settings were numbered from DS1 through DS4, drive (A:) would be set to DS2, and drive (B:) would be set to DS1.

If the settings were numbered from DS0 through DS3 (as they often are), then drive (A:) would be set to DS1 and drive (B:) would be set to DS0. In other words, regardless of how the DS settings are numbered, assign the second setting number to drive (A:) and the first setting number to drive (B:). This is the procedure if a "straight" cable is used. But if a "twisted" cable is used, both drive select jumpers are set to the second-numbered setting. For settings numbered DS1 through 4, the setting for both drives would be DS2. If the settings were numbered DS0 through DS3, both drive select jumpers would be set to DS1. The mystery of straight vs. twisted cables is explained below.

Straight Cable vs. Twisted Cable

The drive control/data cable for floppy drives may be either of two kinds, "straight" or "twisted." In a straight cable, all of the wires (of which there are 34) maintain the same relationship between the controller and the drive connectors. That is to say, pin 1 on the controller is connected (by wire) to pin 1 on the cable center connector, and from there to pin 1 on the cable end connector. The same applies to the remaining 33 pins and wires. Thus, the wiring is said to be "straight through."

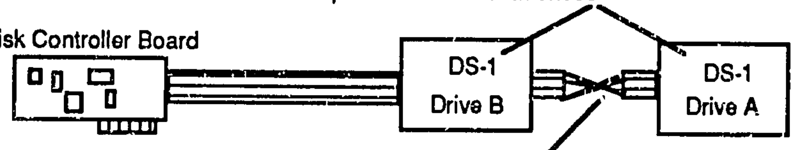
With a twisted cable, leads (wires) 10 through 16 are reversed at a point between the cable center connector and the cable end connector. This effectively reverses the meaning of the names DS1 and DS2 on the drive connected to the end of the cable. If DS1 is selected on one drive, then DS1 must also be selected on the other drive, in order that the drives be selected differently. At first this may appear to be some form of madness, but in fact there is method to it.

IBM created the twisted cable arrangement so that all of the floppy drives could be set to the same selection number before they were shipped out to the world for installation. Or, so that the installer of any third party floppy drives could set both drives jumper selection setting. In theory, one no longer needed to worry about which drive should be set to DS1 and which should be set to DS2.

Figure 13-1 depicts a typical PC floppy drive cable configuration. Note that the drive connected to the end of the cable must have the terminating resistor in place, and that the drive at the center connection of the cable must have the terminating resistor removed. If only one drive is installed, it must have a terminating resistor. The terminating resistor acts as an electrical "load," to prevent signals coming through the cable from being echoed back. Such echoes, if present, would confuse the system.

Some drives use an 8-position switch block rather than a "chip" type terminating resistor. Setting all of the switch positions to "on" serves the same function as having a terminating resistor in place. Setting all of the switches positions to "off" equates to removing the resistance (un-terminates the drive). On drive (A:), all of the switch positions should be set to "on." On drive (B:), all of the switch positions should be set to "off."

Fig. 13-1. A Typical PC Floppy Drive Cable Configuration



Terminating Resistors

To prevent signal echoes from occurring in the drive cable, it is necessary that the last drive on the cable (farthest from the drive controller) have a terminating resistor (TR) installed. The other drive must not have a terminating resistor installed. Note, however, that PS/2 systems do not use terminating resistors on their floppy drives.

The terminating resistor is found on the drive's printed circuit board, which is usually on the top of the drive. On IBM drives, the terminating resistor is labeled "TR." The resistor may be packaged to look like a memory chip or other semiconductor chip. It is usually socketed, and has 16 pins (14 pins on an AT). If it is soldered in place, it will have a jumper labeled "TM." If so, to "un-terminate" the drive, the resistor should not be removed. Instead, remove the TM jumper. Figure 13-2 depicts a typical socketed type terminating resistor and a jumper type terminating resistor.

Figure 13-3 shows the correct floppy Drive Select and Terminating Resistor settings for both straight and twisted cables. Note that for a straight cable, the DS settings for the two drives must be different, but for the twisted cable, the DS settings must be the same, and they must both use the second setting.

Fig. 13-2. Typical Floppy Drive Terminating Resistors

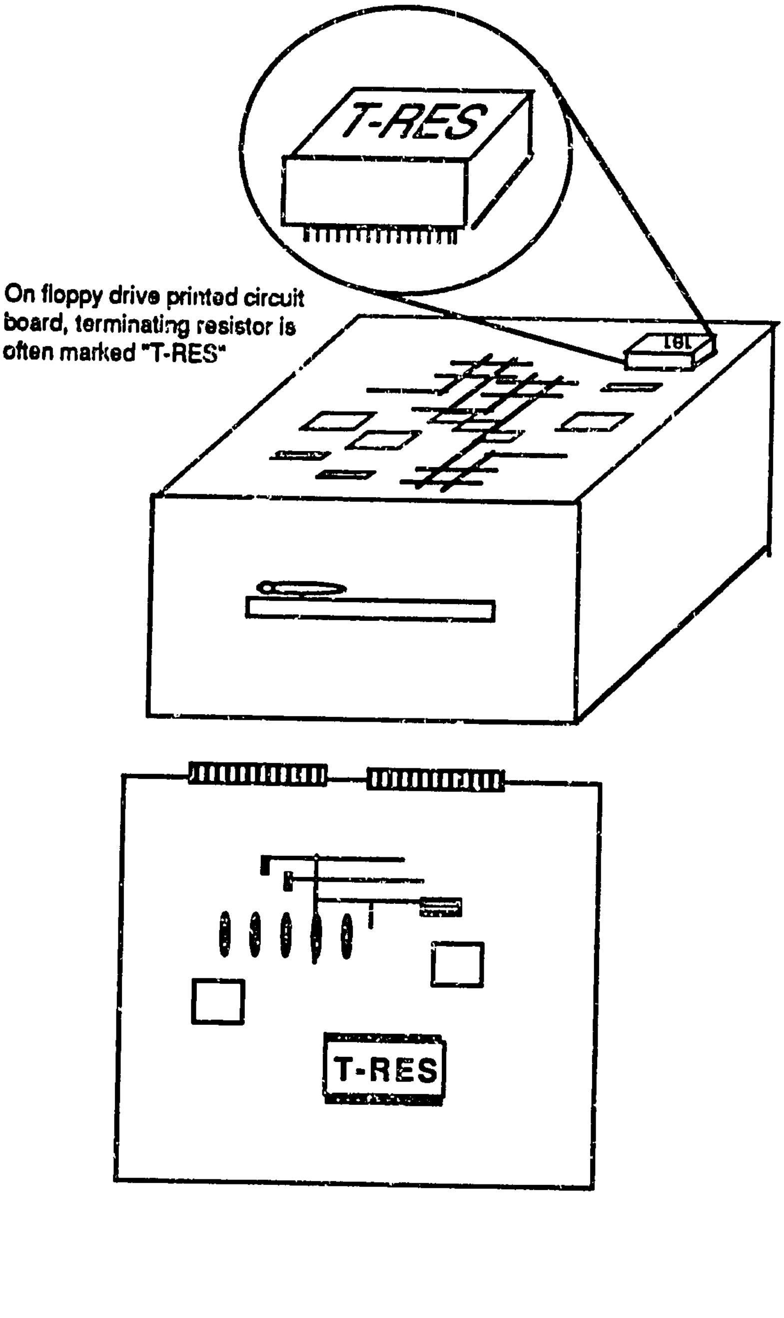


Fig. 13-3. Floppy Drive Select and Terminating Resistor Settings

Cable Type	If Jumpers Are Numbered 0 through 3		If Jumpers Are Numbered 1 through 4	
	Center Connector	End Connector	Center Connector	End Connector
Straight	DS1 = drive (B:) XTR	DS0 = drive (A:) TR	DS2 = drive (B:) XTR	DS1 = drive (A:) TR
Twisted	DS1 = drive (B:) XTR	DS1 = drive (A:) TR	DS2 = drive (B:) XTR	DS2 = drive (A:) TR

* TR = Terminal Resistor; XTR = No Terminal Resistor

Media Sensor Jumper

The media sensor (MS) applies only to 3 1/2 inch, 1.44 MB floppy drives. These drives will allow both 1.44 MB diskettes and 720Kb diskettes to be inserted. If a 720Kb diskette is inserted and the DOS FORMAT command is entered at the keyboard, the media sensor will prevent the system from trying to format the 720Kb diskette to 1.44 MB.

The sensor operates by detecting the presence of a small hole in the diskette jacket. The hole is present only on 1.44Mb, high-capacity diskettes. Note that PS/2 systems do not use drives with the media sensor. On PS/2's, the function is handled by the floppy disk controller that is built into the system board, which allows either 720Kb or 1.44Mb diskettes to be used. The basic issue here involves the level of write current used by the drive, and is beyond the scope of our discussion. The jumper may be recognized by its labeled options: HI; HO; LHI; LHO; OP. To set the jumper, it is necessary to refer to the documentation provided with the drive.

Diskette Changeline/Ready Jumper

In the early evolution of personal computers (prior to the AT), there was a time when everything possible was being done to increase the response speed of the system. One way to increase that speed was to eliminate repeatedly reading the floppy disk to get the same information over and over again. Mainly, it was the disk File Allocation Table (FAT) and the directory that were being read over and over, even when they had not changed. So, systems were designed to read the disk's FAT and directory and store that information in a buffer (in memory). In this way the information could be retrieved much faster, because retrieving information from memory is faster than retrieving from information from a disk. But this created a problem.

If the floppy disk's FAT and directory were read and stored in memory, and then the diskette was removed from the drive and another diskette was inserted into the drive, the system had no way of knowing that the diskette had been changed. Then, when the user commanded the machine to write a file to the diskette, the system assumed that the first diskette (and its FAT and directory) were correct. So, it wrote the file to the second diskette and destroyed whatever else was on that diskette. A lesser problem would occur when the user asked for a directory listing by using the DOS DIR command. The machine would display the directory information from memory rather than reading it from the newly inserted diskette.

The solution to this problem has been to design the computer so that it can sense when a diskette has been removed. This function is accomplished (activated) by setting the diskette change line/ready jumper to "CD." Some machines, in particular the IBM AT, require the changeline function to be activated; other machines may not need it. On the diskette printed circuit board, the jumper will have two choices, "RDY" and "CD." The jumper must be placed across the CD pins (and removed from the RDY pins) to activate the circuit that senses when a diskette has been changed.

Configuring and Installing Hard (Fixed) Drives

Installing a hard drive is not much different than installing a floppy drive. In fact, with a hard drive there are only two configuration settings to be concerned with, the Drive Select (DS) jumper and the Terminating Resistor (TR). Since these items have been addressed for floppy drives in the preceding section, they will not be detailed again here. As with the floppy drives, however, the hard drive cable may be of either the straight-through or the twisted type.

Figure 13-4 shows the correct Drive Select and Terminating Resistor settings for hard drives, for straight cables and twisted cables (both refer here to the control cable of the drive). Note that for a straight cable, the DS settings for the two drives must be different, but for the twisted cable, the DS settings must be the same, and they must both use the second setting.

Also important to note, as shown in Figure 13-4, is that drive (C:) must always have a terminating resistor installed; and drive (D:), if present, must have the terminating resistor removed. If there is no drive (D:), drive (C:) still needs a terminating resistor.

Fig. 13-4. Hard Drive Select and Terminating Resistor Settings*

Cable Type	If Jumpers Are Numbered 0 through 3		If Jumpers Are Numbered 1 through 4	
	Center Connector	End Connector	Center Connector	End Connector
Straight	DS1 = drive (D:) XTR	DS0 = drive (C:) TR	DS2 = drive (D:) XTR	DS1 = drive (C:) TR
Twisted	DS1 = drive (D:) XTR	DS1 = drive (C:) TR	DS2 = drive (D) XTR	DS2 = drive (C:) TR

* TR = Terminal Resistor; XTR = No Terminal Resistor

Hard Drive Connectors and Controller Cables

A typical hard drive has four connectors. They are:

- Interface control connector
- Interface data connector
- DC power connector
- Ground connector

The control, data and power conductors, being multi-line conductors, are "cables." The ground conductor, which is a single conductor, is referred to simply as a lead or wire.

There are usually two ribbon type cables (flat, multi-conductor cables) that connect from the drive controller board to the drive itself. One cable is larger than the other, and it contains 34 lines carrying the drive control signals. The smaller cable contains 20 lines, which carry data to and from the drive. Some drive controllers have an additional 20-line data cable, for connection to an additional drive. A separate cable is provided for the DC power. The ground lead usually has a "tab" type connector by which it connects to the chassis.

Physical Installation

The physical installation of a hard drive is similar to that for a floppy drive, as discussed earlier in this chapter. Attention must be paid to ensure that the drive form factor is correct and that the proper mounting hardware and screws are used.

Of course, all of the cables and the ground lead must be connected before attempting to configure the drive to the system, as discussed later in this chapter. For more information on drive installation, please refer to Chapter 5.

Drive Controllers and Interfaces

A disk drive communicates with the computer by means of controller and interface circuitry, which is usually contained on an adapter board that plugs into one of the system's expansion slots. On AT type machines, the board usually contains the controllers for the floppy disk drive and the hard disk drive.

Strange as it may seem, disk drives and disk controller/interface boards are not manufactured by the same company. And, drives sold by IBM with the IBM label on them are in fact manufactured by outside companies. For IBM XT and AT systems, controllers have been used as shown in Figure 13-5, manufactured by the companies indicated.

Fig. 13-5. IBM XT and AT Hard Disk Controllers

Computer	Controller Manufacturer	Controller Model/Number
IBM XT, 10Mb and 20Mb hard disk	XEBEC Corp.	1210
IBM AT,		
20Mb hard disk	Western Digital Corp.	WD1002 - WA2
30Mb hard disk	Western Digital Corp.	WD1003 - WA2

It is necessary, therefore, to ensure that the controller and the drives are compatible. This is best done by referring to the specifications and requirements published by the controller and drive manufacturers.

Drive controllers are usually designed to support a wide variety of disk drives. Information about the type of drives a controller will support may be stored in a ROM chip on the drive's circuit board. The information thus stored is referred to as the "drive table." Other controllers are accompanied by software containing drive tables, and the tables can be loaded onto the boot track of the hard drive.

Controllers are of several different basic interface designs. Among them are:

- ST-506/412
- ESDI
- SCSI
- IDE

ST-506/412 Controllers

The ST-506/412 nomenclature originated with the Seagate ST-506, 6Mb hard drive, and the Seagate ST-412, 12Mb hard drive. Neither of these drives is still being manufactured. But, the name "ST-506/412" has been carried on as a kind of de facto, interface standard. Thus, one sees reference to a "Standard ST-506/412 interface." An ST-506/412 type controller connects to the disk drive using the 34-lead (pin) control cable and the 20-lead (pin) data cable as described earlier in this chapter. The ST-506/412 controller supports a data density of 17 sectors per track (cylinder). Drive-table data is typically stored in a ROM chip on the drive circuit board.

ESDI Controllers

The ESDI (Enhanced Small Device Interface) controller is a specialized interface for fixed disks and tape drives. It was established as a standard in 1983 by Maxtor Corporation. It is an improvement over the ST-506/412 interface in that the ESDI supports 34 sectors per track (cylinder), which is twice the data density of the ST-506/412. This also allows a much faster data transfer rate, approximately 1024Kb per second, provided the interleave factor on the drive is 1:1. The ESDI is becoming the standard for IBM machines, replacing the older ST-506/412 interface. For example, all PS/2 systems that support a hard drive of 70Mb or greater use the ESDI.

The ESDI controller generally uses a 40 lead (pin) cable for connecting to the hard drive. Certain ESDI models, however, use two cables (34 pin and 20 pin), but these are not interchangeable with ST-506/412 boards. The ESDI allows the drive table to be stored directly on the hard disk. This eliminates the need for storing them on a ROM chip. It is easier to update data that is stored on a disk drive than than it is to replace a ROM chip.

SCSI Interfaces

SCSI (pronounced "scuzzy" by the "techies," i.e., the technicians or computer buffs) stands for Small Computer Systems Interface. Although commonly done, it is technically incorrect to refer to this device as a "controller." The SCSI does not directly interface with disk drives. It is in fact a "general purpose" device that plugs into an adapter slot on the system board, and in turn, allows controllers and other components to be plugged into it. So, even with a SCSI installed, one still needs to have a controller for the disk drives. An advantage of the SCSI is that it has tremendous flexibility and it can greatly expand the number of devices that can be added to the basic system.

The SCSI also offers excellent data transfer rates. With the SCSI-1 (8-bit) standard, data transfer speeds of 32 Megabits per second are attainable. With the new SCSI-2, (16 and 32 bit) standard, rates in excess of 40 Megabits per second will be achieved!

The SCSI has eight ports (called "lunpacs," and numbered 0-7), into which, for example, one may plug CD-ROM (Compact Disk - Read-Only Memory) drives; magnetic disk drives; and communications devices (modems). Actually, only seven ports are available for add-on devices, because one port is used to interface with the computer. Unlike the familiar adapter slot arrangement found in IBM PCs, XTs and ATs, the SCSI is designed to accept add-ons in a "daisy chain" configuration. That is, the SCSI adapter plugs into the computer, and devices are connected as a chain or string, wherein each successively installed device plugs into the one installed before it.

Installing a SCSI device is not a simple matter, especially for retrofitting one on an IBM personal computer. The technical details are beyond the scope of our discussion here. Suffice it to say that the SCSI architecture has been used extensively in Macintosh computers, which do not provide for expansion through adapter card slots as in the IBM type machines.

Because the IBM machines (and clones) have used adapter slots, initial demand for SCSIs in IBM type has been limited. The trend, however, is in the direction of increased SCSI applications, especially with the proliferation of devices that can now can be attached to the personal computer.

IDE Interfaces

The Integrated Drive Electronics (IDE) interface is a low-cost alternative to the ESDI and SCSI interfaces. The economy is achieved essentially by having the bulk of the controller functions built into the hard drive itself, rather than into the interface unit. This makes the IDE interface unit less expensive, but it requires that an IDE drive be used with it. Although the IDE interface may be totally adequate for many applications, its data transfer rate is slow in comparison to SCSI and ESDI interfaces.

351

Configuring the System

In contrast to configuring the disk drive, as discussed in this chapter, there remains the matter of introducing the newly installed drive to the computer system itself. This is necessay so that the drive will be recognized and used by the system. The latter procedure is referred to as "system configuration."

Depending on the model of personal computer involved, system configuration may require setting some switches and jumpers on the system board, or it may be done by using software utilities provided for that purpose. The software, in turn, may require the user to key in data about the options installed, and/or make selections from a menu. For certain machines, the software, once loaded, may do all of the work and reconfigure the system automatically, with no further help from the user.

If a disk drive has simply been replaced with an identical unit, and assuming all of the jumpers, switches and terminating resistors are properly configured, and that all of the cables and the ground leads have been connected, it is not necessary to reconfigure the system. This assumes, of course, that the system was configured correctly before the drive was replaced.

Configuration is required when devices have been removed, added or changed. For systems having a CMOS configuration storage chip, reconfiguration also is necessary when the CMOS battery has failed and is replaced, unless the battery is replaced immediately upon failure, before the chip has lost its data. Further, loss or corruption of the CMOS configuration data, by any cause, requires that the system be reconfigured.

IBM PC and XT System Configuration

ON the IBM PC and XT, configuration is accomplished by setting switches on the system board. The PC has two switch blocks; the XT has only one switch block. Each block contains 8 miniature switches. Figure 13-6 shows the correct switch settings for the IBM PC. The switches to be set are in Block 1, as shown. In the table, the switches are numbered from 1 to 8, as they are on the system board. For the IBM PC, only switch numbers 1, 7 and 8 are used for configuring the floppy drives; switches 2-6 are not used.

In the left-hand column of the table is shown the number of drives that may be installed. Usually this is one or two floppy drives. Do not count any hard drives that may be installed. In the leftmost column of the table, find the number of drives installed. Then read across the row to find the correct switch settings. A "1" in the switch setting column indicates that the switch must be set to its closed or "on" position. To be "on" the switch must be up. To be "off," the switch must be down.

For example, with two floppy drives installed, switch number 1 is set to "off," (down); switch number 7 is set to "off," (down); and switch number 8 is set to "on," (up).

Configuring IBM AT Systems

On an AT system, except for SW1 which selects the default video display, and a jumper that must be set to indicate the amount of memory (RAM) installed, there are no switches to be set on the system board. Configuration is done using the SETUP program found on the diagnostics diskette that comes with the computer. Insert the diskette into the (A:) drive and turn the system on.

If the (A:) drive is one of the items installed prior to the attempt at reconfiguration and the drive does not operate, re-check that the Drive Select jumper is correctly set, the terminating resistor is present on the drive, and all cables and the ground lead are connected. If the drive operates but the configuration is incorrect, a "161" error code will appear on the screen, indicating that the hardware options (e.g., drives, display) have not been selected. Press the F1 function key to run the Setup program. From there, follow the prompts on the screen and enter the data needed.

Fig. 13-6. IBM PC System Board Switch Settings For Floppy Drives

IBM PC	Block 1 Switch Settings*							
Floppy Drives Installed	1	2	3	4	5	6	7	8
0	1	(Switches 2-6 are not used to configure for floppy drives)					1	1
1	0						1	1
2	0						0	1
3	0						1	0
4	0						0	0

* 1 = switch closed (on/up); 0 = switch open (off/down)

Figure 13-7 shows the correct switch setting for the IBM XT. The figure is read the same as was explained for Figure 13-6, above.

Fig. 13-7. IBM XT System Board Switch Settings for Floppy Drives

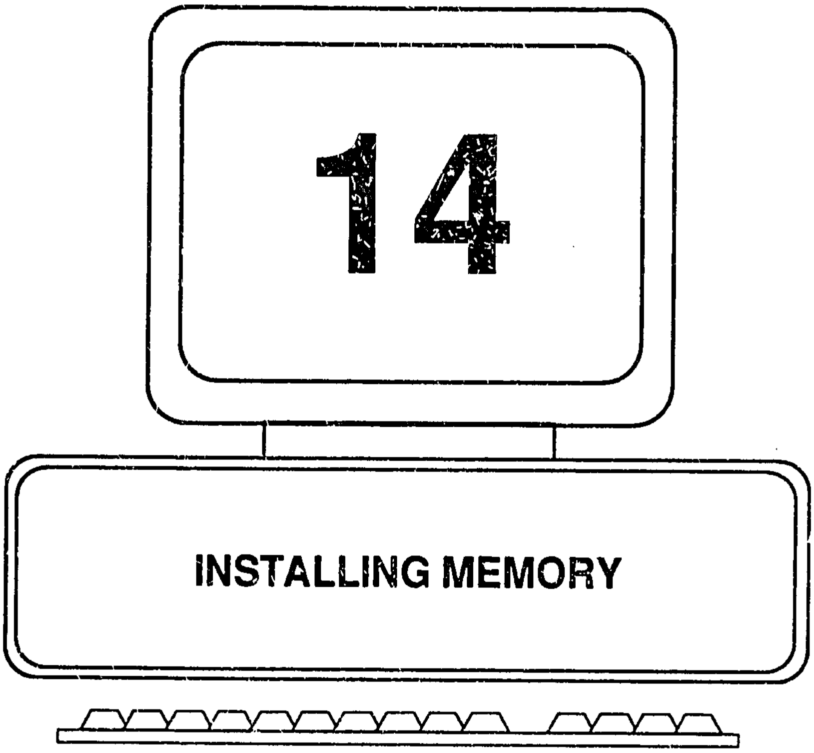
IBM XT	Block 1 Switch Settings*							
Floppy Drives Installed	1	2	3	4	5	6	7	8
0	(Switches 1-6 are not used to configure for floppy drives)						n/a	n/a
1							1	1
2							0	1
3							1	0
4							0	0

* 1 = switch closed (on/up); 0 = switch open (off/down); n/a = non-applicable

Configuring PS/2 Systems

On PS/2 systems, there are no switches to be set on the system board. Configuration is done using the configuration option selected from the main menu presented by the Reference or Starter diskette that comes with the computer. Note, however, that PS/2 Model 25 and 30 need no configuration. The Reference/Starter diskette simply allows the user to confirm that the list of installed options is correct, set the date/time, and prepare the system for moving.

Insert the Reference/Starter diskette into the (A:) drive and turn the system on. If the (A:) drive is one of the items installed prior to this attempt at reconfiguration and the drive does not operate, re-check that the Drive Select jumper is correctly set, the terminating resistor is present on the drive, and all cables and the ground lead are connected. With the Reference/Starter diskette in the (A:) drive, turn the power on. The POST will run, the system will boot up, and then the IBM logo will appear on the screen. Press the ENTER key to continue. The main menu of the Reference/Starter program will appear on the screen. Choose option number 3, "Set Configuration." If the configuration is not correct, the prompt, "Automatically reconfigure system?" will appear. Press "Y" to run the automatic reconfiguration program. This should take care of the reconfiguration. There are other setup options available on the menu, but they will not be covered here.


14
INSTALLING MEMORY

CHAPTER 14: INSTALLING MEMORY

In this chapter, we will see how to find the amount of memory installed on a PC; learn how memory chips are laid out and numbered on a system board; understand how to read memory error messages; and see how all of the components on a system board are numbered. We shall also discuss the proper methods for handling chips and installing memory. Finally, we shall examine the main steps for configuring a system after memory has been installed.

How Much Memory Is Installed?

To begin, it may be helpful to see how we can quickly find out how much user (conventional, base) memory is installed on a machine. Then we will see how and where the memory physically resides on the system board.

To see how much conventional (user) memory is installed, the MEM command is available under DOS 5.0. Type the following command at the DOS prompt:

mem [enter]

Here is the result from the MEM command, for the PC on which this book is being typed:

655360 bytes total conventional memory
654336 bytes available to MS-DOS program
586224 [bytes] largest executable program

As shown above in boldface type, the machine has a total of 655360 bytes of conventional memory installed. Since RAM is more commonly expressed in Kilobyte units, we may find that value by dividing the total bytes shown by 1K byte (which is 1024 bytes):

655,360 ÷ 1024 = 640K bytes of conventional memory installed

Alternatively, the DOS CHKDSK (for "checkdisk") command can be used. It will report the capacity of the current (default) drive, and also the amount of conventional memory installed. At the DOS prompt, enter:

chkdsk [enter]

Here is the result from the CHKDSK command, for the PC on which this book is being typed:

Volume ACR 120 Created 01-04-1992 11:09a
Volume Serial Number is 281C-128C

120315904	bytes total disk space
73728	bytes 2 hidden files
38912	bytes in 15 directories
17098752	bytes in 583 user files
103104512	bytes available on disk
2048	bytes in each allocation unit
58748	allocation units on disk
50344	available allocation units on disk
655360	**total bytes memory**
586224	bytes free

As shown above in boldface type, the machine has a total of 655360 bytes (640K bytes) of conventional memory installed, the same amount as found using the MEM command. Since part of the total memory is presently occupied by DOS and drivers for the various devices on the system, the amount of memory still available or "free" (586224 bytes, or about 542K bytes free) is less than the total memory shown.

The first eight lines of the above CHKDSK result pertain to the hard disk (C:), and hard disk storage capacity has nothing whatever to do with the amount of memory installed. For the reader who may be interested, the entries are explained as follows. This machine happens to have a 120Mb hard drive. When the disk was formatted, it was given a volume name, as shown, which just happens to be the owner's initials and the size of the hard drive (Mb). The date and time are included automatically. The two hidden files are the DOS system files, IBMBIO.COM and IBMDOS.COM (the machine is running under IBM DOS 5.0). There are 15 directories (occupying 38912 bytes for their structure and housekeeping functions, not the files stored in the directories). There are also 583 user files (programs and documents) presently occupying 17098752 bytes (or, about 16 Mb) of disk space.

This particular hard disk is a non-IBM device. It happens to allocate disk space in units of 2048 bytes, which is four times the 512-byte sector value most commonly seen: 512 bytes x 4 = 2048 bytes (2Kb).

The conventional memory on which we have just checked is provided physically by semiconductor "memory" (RAM) chips. The chips may be mounted on the system board (motherboard), on a memory expansion card (in addition to system board memory) or, in certain IBM PS/2 Models, on a special, plug-in memory board called a SIMM (Single In-line Memory Module).

There are generally two occasions when opening up the PC and removing and installing chips is required. One is in replacing faulty chips; the other is in installing additional memory to the system. To do either of these tasks, it is helpful to understand something about how the chips function, how they can be identified, and how they are laid out (located) within the system.

It is also necessary to be able to interpret "memory error" messages, and from the messages to be able to locate the exact chip involved. Finally, it is important to know how to handle chips, so that they can be removed and replaced correctly, without damaging them. These matters are addressed below.

Memory Chip Size Specifications and Functions

A memory chip is described, in part, in terms of its "depth" and its "width." These terms refer not to the physical dimensions of the chip. The depth of a chip is the chip's storage capacity measured in bits. The width of a chip describes the number of bits that can be accessed from the chip in a single operation. For example, a 64K x 1 bit chip is 64K bits "deep" (it stores 64K bits of information) and it is 1 bit wide (it can accept or give out only one bit of information at a time). Depth and width are explained further, below.

The size (storage capacity, or "depth") of a memory chip is expressed in bits. Most commonly it is expressed in units of 1,024 bits, or K bits. Thus, a 64K chip can hold 64 x 1024 = 65,536 bits of information.

Recall from Chapters 6 and 8 that a bit is a binary digit, which can have only one of two values – either 1 or 0. Recall also that it takes 8 bits to equal one byte, and one byte is the amount of binary code used to represent a character, such as the letter "A." Of course, the binary number equates to a decimal number and the decimal number is the "ID" number for the character. But as far as the memory chip is concerned, it simply stores "bits," and all of the rest is left to other and "smarter" elements of the system.

Memory Chip Layout on the System Board

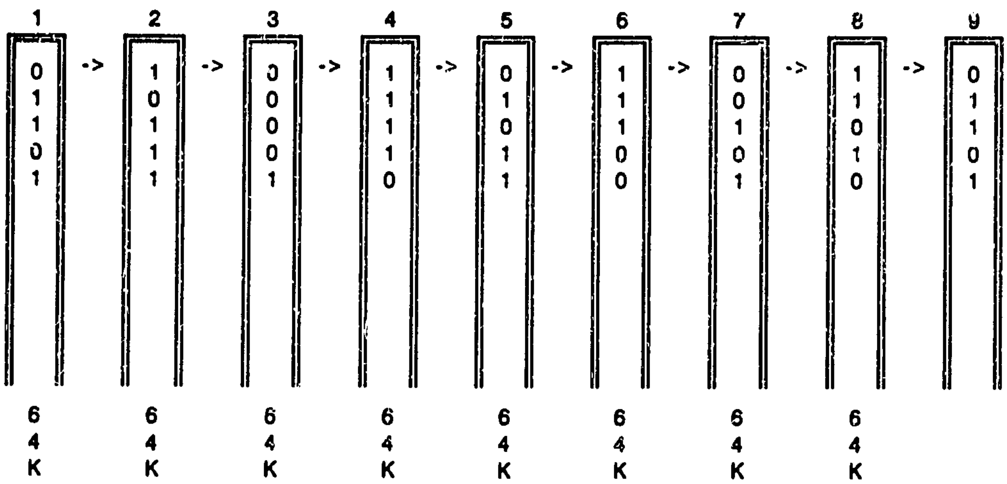
As we shall see momentarily, memory chips are (normally) arranged in a rectangular pattern of rows and columns on the system board. This makes them easy to recognize. As it happens, it takes 8 chips to represent one byte of information. Why should this be, considering that any one chip can store many thousands of bits, and it takes only 8 bits to form one byte? The answer is that while a chip can indeed store thousands of bits at a time, only one bit can be read from a chip at any one time. Thus, to represent (store) one byte, the 8 bits of the byte must be stored one each, across 8 chips in a row. Thus the 8 bits of any particular byte of data exist in 8 different memory chips at a given time. As always, there are exceptions, and they are addressed further on. For the moment, however, we will assume that the chips can be read only one bit at a time. Such chips are said to be "1 bit wide."

The width of a chip, defined in bits, tells how many bits can be read from the chip simultaneously. A chip that can be read only one bit at a time is "1 bit wide." Certain chips are 2 bits wide, and other are 4 bits wide. A 2-bit wide chip can receive and give out two bits worth of data at a time. Thus, as one might expect, it takes only four such chips to store 1 byte (8 bits) of data. So, one would expect to see these chips arranged in four columns (not counting any additional chips used for parity, as will be explained in a moment) rather than the 8 columns as discussed for chips that are 1 bit wide.

If the chips are 4 bits wide, it takes only two of them to store one byte of data. So, here one might find the chips arranged in just two columns (not counting any additional chips used for parity, explained below).

Figure 14-1 shows a row of nine, 64K x 1 bit wide chips. The first 8 chips are for storing data. The ninth chip is used for checking the data in the other 8 chips. The process is called "Parity checking." In Figure 14-1, the chips are shown "standing on end," with the binary digits (bits) stacked up inside each one. This is a conceptualized view, not intended to construe the actual inner structure of the chip. As shown, each chip can store up to 65,536 (64K) bits, each bit being represented by a 0 or a 1. To represent (store) a byte of data (8 bits), for example, the binary number 01010101, each of the 8 chips stores one of the 8 bits, as shown in the first row of Figure 14-1. The arrows in the figure indicate that the bits are to be read across the row, from left to right. The PC, however, can access the 8 chips simultaneously, and thus write or read an entire byte of memory data in a single operation. In the arrangement shown, we may refer to the 8 chips in the row as being "associated." As we shall see later, on some boards the associated chips are arranged vertically, in columns instead of in rows.

Fig. 14-1. Conceptual View of Memory Chip Function



Parity Checking

The ninth chip, shown at the right-hand column of Figure 14-1, is not used to store data. It is used in an error checking procedure called "parity checking," as follows. When data is written across the row of the other 8 chips (each of which may thus be set to 1 or 0), the number of ones in the row is added up. If that number is an odd number, the value of the ninth chip is set to 1. If the number is even, the value of the ninth chips is set to 0. Later, when the 8 values are read from the data chips, the number of ones read is again added up and found to be either odd or even. This result is compared to the value stored in the ninth chip. If the values agree, it is assumed (for purposes of this test, anyway) that what was read is what was written earlier. If the value in any one of the 8 data chips had changed for any reason, (e.g., a "0" had changed to a "1," or a "1" had changed to a "0,") the test would be failed, and a "parity" error message would be displayed on the monitor.

Of course, if two bits had somehow changed, say, that a 0 in chip number 1 had become a 1, and 1 in chip 5 had become a 0, the parity test would have no way of detecting it, because the total number of ones and zeros across the row would not have changed. But, the probabilities of this happening are so low as to still make the parity test well worth having. Note that certain " inexpensive" computers save money by eliminating the parity chips, and thus the parity test. These machines will have 8 chips in a row, rather than nine chips as we have been discussing.

We can now see why it is convenient to have the memory chips arranged in rows and columns. It would be even more convenient if a chip error message told us something to the effect that, "The faulty chip is in row three, column 2." In fact, we are told the chip's location, more or less, but the numbering of the rows and columns is just a bit more complicated than that. One reason for this "complication" is that the associated chips as we discussed earlier are on some boards arranged in a row, while on other boards they are arranged in a column. Furthermore, the groups of associated chips are referred to as "banks" of chips, and a bank could consist of more than a single row (or column) of chips. Fortunately, the rows (or columns) that make up a given bank are always located adjacent to each other. Let's take a look at how some of the more commonly encountered boards are laid out. As each diagram is described, take note of how the chips, rows, banks and columns are numbered. These numbers are used in the error codes (covered later) to identify the location of a faulty chip.

Typical Memory Chip Layouts

Figure 14-2 shows the layout of memory chips for the IBM PC (16K chips, 64K total on board). The chips are arranged in four rows (banks), numbered 0-3. Bank 0 is toward the back of the machine. The corresponding error code numbers as shown in parentheses on the diagram are 00, 04, 08 and 0C. When any of the characters A-F appear in memory row/bank numbers, they are hexadecimal numbers. Recall from Chapter 8 that hexadecimal "A" is 10 decimal; "B" is 11 decimal; and "C" is 12 decimal. So, the mysterious "C" in bank "0C" is just the hexadecimal form of the decimal number "12." The rows might have just a well been numbered in decimal as 00, 04, 08 and 12. The columns of chips are numbered P (for parity) and 0-7 for the data chips. The error code numbers across the row are 00, 01, 02, 04, 08, 10, 20, 40 and 80. We will deal with the error code numbers momentarily. Note that the space between the first and second columns is wider than that between the remaining columns. On the IBM PC/XT (Figure 14-3), the columns are equally spaced. Otherwise, the layout is the same, with four rows (banks) and nine columns of chips.

Figure 14-3 shows the PC/XT board as having four banks of 64K chips, for a total of 256K on board. The chips are arranged in four rows, and nine columns, at the left-front area of the system board. Each row is a "bank," and the banks are numbered from 0 to 3, starting from the top row. (Bank 0 is toward the rear of the machine; bank 3 is toward the front.) The corresponding error code numbers as shown in parentheses on the diagram are 00, 04, 08 and 0C. (0C is a hexadecimal number; its decimal equivalent is 12.) The first column (at the left) of chips is labeled "P," for "parity." The remaining 8 chips are numbered consecutively, from 0 through 7. The error code numbers across the row are 00, 01, 02, 04, 08, 10, 20, 40 and 80.

Fig. 14-2. IBM PC (16K/64K) Memory Chip Layout

BANK	P	0	1	2	3	4	5	6	7	
0 (00)	00	01	02	04	08	10	20	40	80	(16K)
1 (04)	00	01	02	04	08	10	20	40	80	(16K)
2 (08)	00	01 x	02	04	08	10	20	40	80	(16K)
3 (0C)	00	01	02	04	08	10	20	40	80	(16K)

Four banks of 16K each = 64K

Fig. 14-3. IBM PC/XT (64K/256K) Memory Chip Layout

BANK	P	0	1	2	3	4	5	6	7	
0 (00)	00	01	02	04	08	10	20	40	80	(64K)
1 (04)	00	01	02	04	08	10	20	40	80	(64K)
2 (08)	00	01	02	04	08	10	20	40	80 *	(64K)
3 (0C)	00	01	02	04	08	10	20	40	80	(64K)

Four banks of 64K each = 256K

Figure 14-4 illustrates the RAM layout for the 640K IBM PC/XT system board. The physical layout is the same as for the 256K PC/XT.

Fig. 14-4. IBM PC/XT (640K) Memory Chip Layout

BANK	P	0	1	2	3	4	5	6	7	
0 (00)	00	01	02	04	08	10	20	40	80	(256K)
1 (04)	00	01	02	04	08	10	20	40	80	(256K)
2 (08)	00	01	02	04	08	10	20	40	80	(64K)
3 (0C)	00	01	02	04	08	10	20	40	80	(64K)

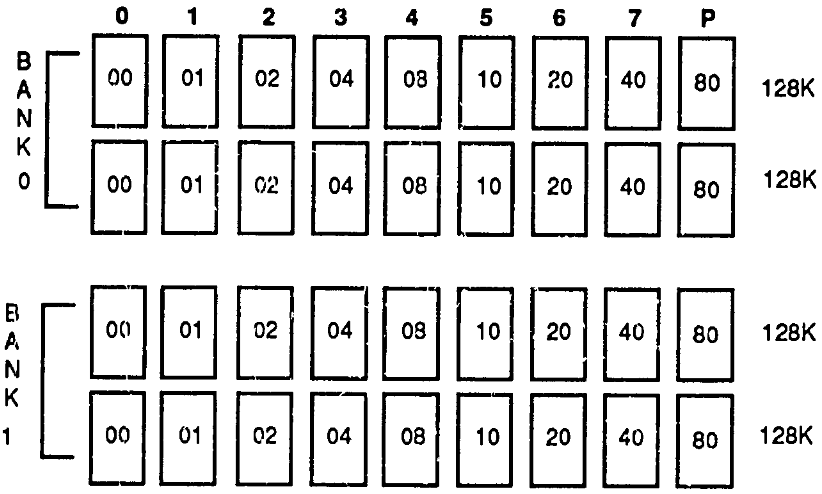
Two banks of 256K each, plus two banks of 64K each = 640K

For the AT, there are two different types of motherboards, with both types supporting 512K bytes of RAM. The early ATs had a "Type 1" board, containing thirty-six, 128K x 1 bit chips. And, these boards had the same horizontal (row) arrangement defining their banks as for the PC and PC/XT. A later version of the AT motherboard (Type 2) contained eighteen, 256K x 1 bit chips, and the banks were oriented vertically (in columns, rather than rows.)

Figure 14-5 depicts the arrangement of memory chips for the IBM AT, Type 1 motherboard. As with the PC and the PC/XT, there are four rows and nine columns of chips, but on the AT Type 1 board, the parity chips make up the right-hand column, and once again they are labeled "P," for parity. The 8 data chips are numbered from 0 through 7, beginning at the left. Whereas on the PC and PC/XT a single row of chips constitutes a bank, on the AT Type 1, two rows of chips form a bank.

The significance of the banks will become clear later, when we cover the subject of error codes. Essentially the size of a bank is determined by the width of the system bus. An 8-bit bus can access 8 chips at a time, so a memory bank in an

Fig. 14-5. IBM AT Type 1 Memory Chip Layout



Two banks, 0 and 1. Each bank contains 256K, for a toal of 512 K on the board

8-bit bus system also has 8 bits (chips). But, a system having a 16-bit bus can access 16 bits. Therefore, it takes two rows of 8 bits (two rows of chips that are 1 bit wide) to form a 16 bit bank.

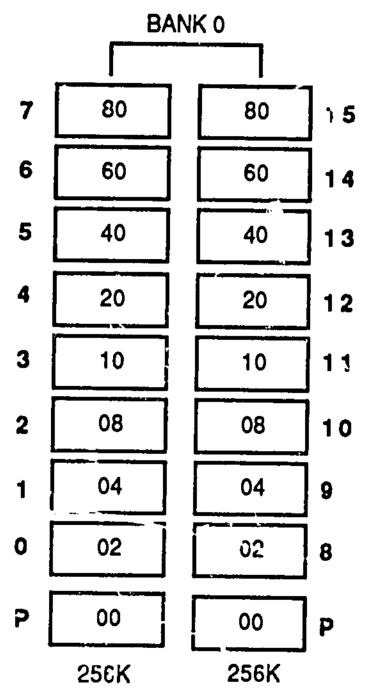
Remember not to confuse the capacity of a chip with its width. To the system bus, a chip that is 1 bit wide can provide only one bit of data at a time. A 32 bit bus, for example, can access one bit from each of 32 different chips that are 1 bit wide; and it does not matter whether the chips are 64K, 256K or 1024K in capacity. Accordingly, the chips are grouped into banks to accommodate the capacity of the bus.

Further, on the AT Type 1, the numbering of the banks begins at the front of the board (bottom of the diagram, as shown). The two front rows form Bank 0, and the next two rows toward the rear of the machine form Bank 1. Also shown, to the right of the memory chip banks, are three jumpers, labeled J18, J19 AND J20.

If only 256K bytes of system board memory is installed, pins 2 and 3 on J18 must be connected (jumped). However, if 512K bytes of memory is installed, the jumper on J18 must instead be across pins 1 and 2. If the J18 jumper is not set correctly, the system will not recognize the correct amount of memory installed.

Figure 14-6 show the layout for an AT Type 2 motherboard (i.e., the AT Model 339). The board has 18 RAM chips, each being a 256K x 1 bit chip. The 18 chips are arranged in two columns of nine chips each, and both columns together form a single, 16-bit bank (which is Bank 0.)

Fig. 14-6. IBM AT Type 2 Memory Chip Layout



Organized as a single bank; total of 512 K on the board. (Chips "00" are for parity)

Chip Location Numbers and Error Codes

We suggested earlier that, ideally, a chip should be located simply according to the row and column it occupies. If the rows were numbered 1,2,3..., and columns were lettered A,B,C..., then a given chip could be identified by its row and column, e.g., A3 or B4 or C1. But, the rows and columns are not numbered as simply as that, and the numbering scheme varies somewhat from one type of board to another. The basic idea of chip location identification by its row and column, however, still applies in general. The location of a chip is given by two, two-digit numbers, xx and yy. The first number (xx) identifies the bank in which the chip resides. The second number (yy) identifies the specific chip in the bank (or close to it). Thus, a chip might be identified as 0801. Where is it on the board? The first two digits, 08, tell us the bank number, and the second two digits, 01, tell us the chip number.

Of course, we must be dealing with a specific board and its particular numbering arrangement. Here we will deal first with an example for the IBM PC, and then for the PC/XT. Referring back to the diagram of the board for the IBM PC (Figure 14-2), we see that the four banks are numbered 00, 04, 08, and 0C. The chips in each row are numbered from left to right as 00, 01, 02, 04, 10, 20, 40 and 80. We are looking for chip 0801.

We split the number 0801 into its two-digit parts: 08-01. Next, we find bank 08; and within that bank we find chip 01. On the diagram in Figure 14-2, chip 0801 is marked by an "x." Success!

If the chip just mentioned had been faulty, an error message from the POST (Power-On Self-Test or some other memory test, perhaps run from a software package) would have displayed a memory error code of 0801. The message would also say "Parity 1 error," because the chip is installed on the motherboard. If the bad chip were on a memory expansion card, the message would say "Parity 2 error." That's how you can tell the difference between faulty motherboard chips and RAM chips in other places: "Parity 1" means the bad chip is installed on the system board. For later IBM machines, the chip location numbers and error messages work pretty much the same as we have just seen. But, there are three extra zeros into the location number since the days of the PC. So, for a PC/XT, the error code we just solved would be: 08 000 01.

Suppose you were running an IBM PC/XT, 640K machine with a motherboard like the one shown in Figure 14-3, and you received the following error message:

Parity 1 error 0800080

Can you find the faulty chip? The answer is easy. Just write down the first two digits and the last two digits, and ignore the middle three zeros. The first two digits of the code are 08, and the last two digits are 80. So, the bad chip is in bank 08, and the chip number is 80. In Figure 14-3, it is marked with a (*).

349

How Else to Find a Bad Memory Chip

Of course, a bad chip may be identified by way of an error code message, as we have just discussed. And, one should always be alert to any recall notices issued by the chip manufacturer. If repeated problems are encountered with a specific make and series of chip, it would be well to contact the manufacturer, to see whether the chip design was defective and whether the chip has been recalled. Usually, this will entitle the owner (purchaser of record) of the machine to a free replacement chip.

But sometimes, a chip will work part of the time and not others – it will have an "intermittent" fault. This problem may be related to the operating temperature. When it is cold, the chip may work fine, but after warming up, it may become unreliable.

The opposite is also possible. Another problem is that the chip may have crept slightly out of its socket, such that its electrical contact with the board is uncertain. If new chips were just installed, they may have been put in backwards, or a pin may have become bent or broken in the process. Note that if a chip has been installed backward and the machine has been powered up, the chances are good that the chip will be ruined.

To map the memory chips on the system board, you can use a chip that is known to be faulty. Replace each existing chip on the board one at a time, and for each replacement run the POST (Power-On Self-Test) by turning the computer power off and then, after 10-15 seconds, turning it on again. The memory error message displayed by the POST will identify the faulty chip. You then will know the location code number for the physical position of the faulty chip. This may seem to be tedious task, but it is a sure way of mapping the memory chip layout.

As far as temperature is concerned, a chip operating properly should feel warm to the touch, but not decidedly hot, and not cold. If you are having trouble locating a faulty chip, let the system warm up for a while (perhaps 10 minutes or more), and then feel the chips, Compare them for temperature. If one feels noticeably warmer or cooler that the others, it may be the culprit.

Be sure all of the chips are installed in the correct orientation – with the notch (or little dot which serves the same purpose as a notch) on the chip facing the same way as the other chips installed. Be sure all of the chips are firmly seated in their sockets – press down firmly on each one.

Finally, to diagnose an intermittent problem may require repeated testing over an extended period of time. A third party diagnostic such as Checkit may be the answer here.

Identifying Types of Memory Chips

Above, we have spoken of the capacity and the width of a memory chip. But there is more to it than that. Memory chips are designed to operate at or below a certain rated speed. The speed of a chip is measured in nanoseconds. A nanosecond is one billionth of a second (.000000001 second).

Personal computer memory chips range in speed from about 250 nanoseconds (slow chips) down to 53 nanoseconds (very fast). In replacing a chip, the new chip must be equal to or greater in speed than the chip it replaces. A chip with a slower-rated speed will not work, unless the chip being replaced was of a speed higher than the system needed in the first place. Chips having a speed greater than is needed by the system will operate, but the higher speed chips cost more, and the system will not take advantage of their full speed.

Information Printed on Memory Chips

The following information is typically (but not always) found printed on a memory chip (and other types of chips, as well).

- Manufacturer's part number
- Manufacturer's logo
- Country where manufactured
- Date of manufacture
- A number indicating the chip type, capacity and speed

The number indicating the chip type, capacity and speed naturally has three components. The first tells the type of chip. For a memory chip, the digits usually are "41." The next sequence of digits (usually from 64 to 1024) tells the chip capacity (depth) in K bits (usually). The final digits, normally separated from the preceding digits by a dash (hyphen), indicate the speed of the chip in nanoseconds. As an example, the number 41256-15 on chip would indicate that the chip is a memory chip (41), having a capacity (depth) of 256K bits, and a speed of 150 nanoseconds. Oops! The last two digits are 15, not 150. So, why not a speed of 15 nanoseconds? The trick is to recognize that commonly found dynamic memory chips for personal computers (explained below) do not operate in the 10-

30 nanosecond range. So, when the last two digits are -10, -15, -20, and -25, multiply by 10 to find the true speed.

Dynamic Random Access Memory (DRAM) is the type of chip used in conventional memory banks. It is called "dynamic" because it must be refreshed by the system frequently and at regular intervals, or else it loses its data. This is of no particular concern to the user, and it requires no action from the user. DRAM is less expensive and much slower than Static Random Access Memory (SRAM). SRAM is high-speed, "intelligent" RAM. Because of its speed, SRAM is commonly used for "RAM caching." RAM caching is a process where frequently used information stored in DRAM is read into and accessed from the SRAM cache. By using the faster SRAM in this manner, the machine can operate faster – it eliminates some of the repetitive accesses to the slower DRAM storage.

A similar process is used, where information stored in a ROM chip (which is slower than a DRAM chip) is copied into DRAM. The ROM information thus can be accessed from the DRAM much faster than if it had to be retrieved each time from the ROM chip. The DRAM that stores the ROM information is referred to as "shadow RAM."

To add further confusion, some manufacturers use digits other than "41" to indicate a memory chip. These other digits are 37, 42 and 66. Even worse, some manufacturers may "add 1" to the memory capacity normally shown. Thus, the usual 64 (K bits) may be indicated by "65." An hypothetical example of what was just stated would be: 3765-25. This is a memory chip (37) having 64K bits capacity (the 65 = 64K bits), and a speed of 250 nanoseconds (25 x 10 = 250).

As mentioned earlier, there is no standard way of marking chips. Sometimes, the chip characteristics are combined with other letters or digits to form a combination code number which becomes both the manufacturer's part number and the chip's capacity and width indicators. The number may, for example, begin with some letter characters. For example, an hypothetical chip marked AB1256-15, the "AB" has to do with the part number; next, the "1", indicates the chip width, which is "1 bit." The digits "256" indicate the capacity (depth) of the chip, which is 256K bits. The "-15" indicates that the speed of the chip is 150 nanoseconds. Figure 14-7 lists the most common configurations of chip depth and width.

Note: Memory chips are commonly constructed to have two rows of pins by which they connect to the motherboard. Replacing the word "two" with "dual," and the word "row" with "in-line," the chips become "Dual In-line Pin," or DIP chips. Thus, conventional memory is also referred to as "DIP" memory.

Fig. 14-7. Common Memory Chip Width and Depth Combinations

Digits Printed on the Chip	Chip Width (Bits)	Chip Depth (Capacity, K bits)
164 (read as 1-64)	1	64
264 (read as 2-64)	2	64
464 (read as 4-64)	4	64
1128 (read as 1-128)	1	128
2128 (read as 2-128)	2	128
4128 (read as 4-128)	4	128
1256 (read as 1-256)	1	256
2256 (read as 2-256)	2	256
4456 (read as 4-256)	4	256

Single In-Line Memory Modules (SIMMS)

Later model personal computers, especially the IBM PS/2 series, are making more and more use of SIMMS in place of individual memory chips. A SIMM is a "Single In-Line Memory Module." It serves the same purpose as the individual chips as discussed in the preceding paragraphs. A SIMM, however, consists essentially of a small circuit board with memory chips soldered to it.

A SIMM occupies less physical space and can hold more data than the equivalent capacity configured as individual chips. Moreover, SIMMS are designed to hold entire bytes of data as opposed to the single-bit storage discussed earlier for the boards using individual chips. This greatly simplifies the problem of locating a faulty unit. But, it also means more expense, because a faulty chip on a SIMM requires that the entire SIMM be replaced.

Numbering Scheme for System Board Components

While we are on the subject of chip identification, the following additional information concerning the numbering of components may be helpful.

- The components on a system board (motherboard) each have an identification code on them. The first part of the code indicates the type of component, as follows:

357

C	=	Capacitor	R	=	Resistor
D	=	Diode	SW	=	Switch
J	=	Jumper	T	=	Transformer
L	=	Coil	U	=	Integrated circuit
Q	=	Transistor	Y	=	Crystal

- The second part of the code refers to the number of the component, where the like components are numbered in sequence beginning at the upper left (toward the rear of the system) corner and counting from left to right. The components will not be arranged in neat rows, but they are generally numbered row by row, to the extent that their layout approximates rows. For example, if component U4 were located at the right hand end of the board, then the next "U" component, i.e., U5, would be in the next "row" (at least it would be more toward the bottom of the board (nearer the front of the system).

Figure 14-8 illustrates the general idea. It is not to suggest that components are in the locations shown. Rather, it illustrates that there is a pattern to numbering the components, which makes them easier to find and identify. In the figure, notice how the number for each type of component increases as the location of the component moves from left to right, and downward toward the bottom of the board (front of the system). The Integrated Circuit (U) components here are shaded to make them easier to spot.

Fig. 14-8. Hypothetical Layout Illustrating Component Numbering Scheme

Rear of system (Top of Board)					
J1			D1	Y1	
	C1	U1	U2	R1	C2
		C3	R2		
	U3			D2	U4
U5	R3	C4		U6	U7
				J2	J3
Front of System (Bottom of Board)					

Handling and Installing Memory Chips

The basic procedures for handling memory chips apply to all semiconductor chips. All are fragile devices, vulnerable to rough handling, heat, moisture, dust/dirt, and static electricity.

Wearing Proper Clothing and Minimizing Electrostatic Discharge

No doubt most of us have experienced that irritating electrical shock, after walking across a carpet and touching a doorknob, or some variation of this situation. Although the electrical current involved in such discharges is quite low, the voltages can be very high. A discharge that can barely be felt has to be about 3000 volts or more. Water in the air, in the form of humidity, reduces electrostatic buildup. As the humidity becomes lower, which usually happens when air temperatures decline, the potential for electrostatic discharge problems increases.

Many materials, such as nylon, wool, silk and fur, are prone to electrostatic discharge. Human hair and skin are also good sources of electrostatic discharge. The molecular structure of the foregoing substances is such that when they rub against other materials, they lose electrons, which creates an electron deficit or shortage. The deficit is maintained until contact is made with some other material, such as metal, which gives up electrons freely. On contact, the electrons "jump" (discharge) from the metal to the material having the electron shortage. These discharges may represent many thousands of volts. Since it takes only about a 1000-volt discharge to damage a chip, is easy to see why the following precautions must be taken:

- Avoid wearing clothing of the type of materials mentioned above.

- Avoid creating an electron deficit by not shuffling across the carpet, and by not handling materials or fabrics so as to create friction between them.

- Avoid wearing leather-soled shoes when working on a PC. The shoes insulate you from the floor, and thus help to retain the potential for an electrostatic discharge between you to the machine or any components that you may touch. Working in your stocking feet is a good idea, if it is permissible.

- Before touching the PC or any components, whether or not they are loose or installed on the machine, first discharge any static electricity by touching a metal object or a ground point on the PC chassis.

- Handle printed circuit boards by grasping them first by a grounding point on the board.

- Have the main power switch of the PC turned off, but leave the power cord connected to the wall outlet. This will provide an electrical ground to the chassis.

Handling Chips Carefully

The physical handling of chips (and, indeed, any component of a PC) should be done with reasonable care and gentleness. Avoid grasping, placing or storing chips in any way that would twist or compress the body of the chip, or which would bend any of the pins.

Avoid storing chips in hot areas (such as in a car in summertime) or where they are exposed to any external source of heat, including direct sunlight.

Any object that is cooled (e.g., from being in an air conditioned vehicle or room) and then carried into a warmer, humid area, is subject to condensation as well as to thermal expansion. If such temperature and humidity change is experienced, allow the component time to warm up to room temperature, and be sure to wipe off any moisture, using a clean, dry cloth.

Removing a Chip

It is simple to remove a socketed memory chip from a system board. The chip is held in the socket only by means of the pins on the bottom of the chip.

Note: Attempting to remove and replace chips that are soldered to a circuit board is not advised. Soldering iron or "guns" produce a lot of heat and the heat will likely destroy the chip.

To remove a socketed chip, insert a thin blade carefully under the edge of the chip and pry the chip up slightly (not all the way out). Repeat the prying action at several more points around the bottom of the chip, gently "walking" the chip out of the socket. Never attempt to simply grasp a chip with your fingers and pull it out of its socket. That approach will invariably damage the chip.

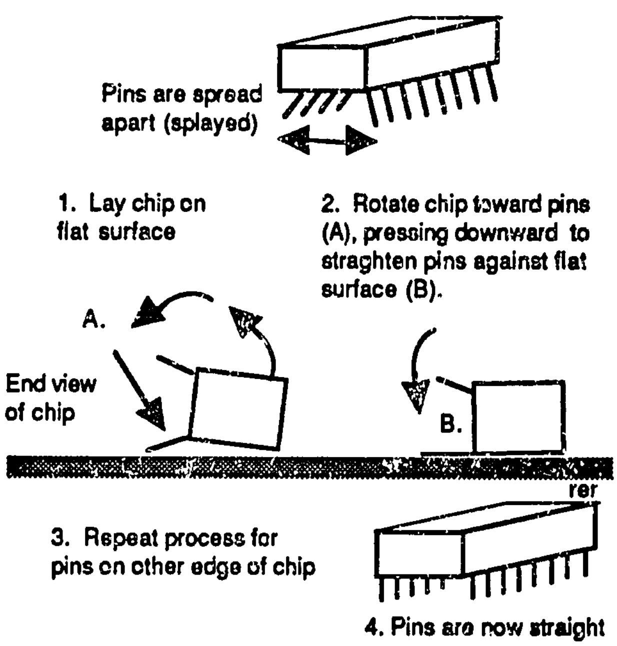
An alternative is to remove the chip by using a chip extractor tool. The tool is a tweezers-like device, which grasps the opposite ends of the chip so that the chip can be pulled out. Before using this tool on a "live" system, it is highly recommended that much practice be done using a defunct board, where damaging chips during the learning process will not matter.

Installing a Chip

When a new chip arrives, its two rows of pins may be splayed (bent outward). Do not attempt to install the chip while the pins are in that condition, and do not attempt to straighten the pins one at a time. Instead, lay the chip on its side, on a flat surface, as illustrated in Figure 14-9. Apply pressure to the chip body, rotating the chip slightly toward the pins, while keeping the bottom row of pins in contact with the flat surface. Inspect the chip to see that the first row of pins are now vertical with respect to the body of the chip. Repeat the procedure for the second row of pins.

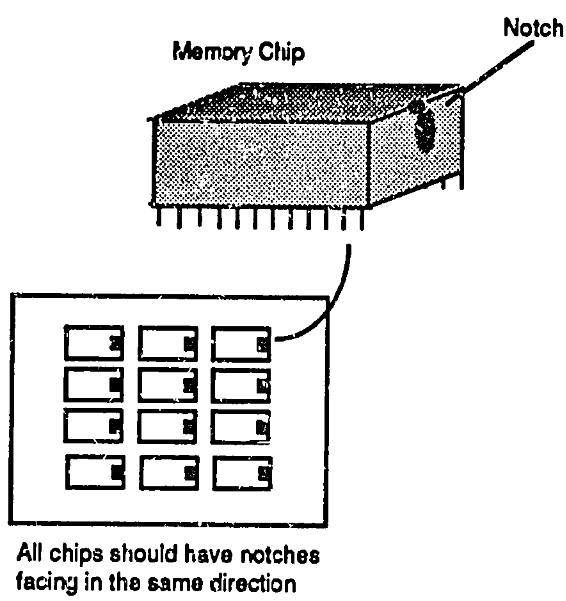
To install a chip, it must be oriented properly (i.e., facing in the correct direction on the circuit board). A chip can be physically inserted two ways. If it is inserted with the wrong orientation, it will be destroyed when the power is turned on.

Fig. 14-9. Straightening the Pins on a Memory Chip



First locate the chip's "orientation notch." One end of the chip will have either a small notch or a marking that simulates the notch found on most chips, as shown in Figure 14-10.

Fig. 14-10. Location of RAM Chip Orientation Notch



Looking at a chip from above, the pins are numbered counterclockwise, going around the chip and ending up with the highest numbered pin opposite pin number 1. The orientation notch is next to pin number 1.

In general, system boards are designed so that all of the notches on the installed chips are facing the same direction. So, the easiest approach is to orient the notch on the replacement chip in the same direction as the other chips on the board.

If the notch orientation for the chips on the board is unclear, there is another for finding pin number 1. This requires examining the underside of the circuit board. In so doing, it will be seen that each pin of the chip socket has a small spot of solder holding it in place. These spots are called "solder pads." Most of them are round, but the pad for pin number 1 will be square. Find the square pad, and turn the board upright, noting the location of pin number 1. Orient the chip so the notch is next to pin number 1.

With the chip correctly oriented as explained above, align the pins to the socket holes and press the chip straight down into the socket. Certain chips may require a bit of pressure in order to seat them. Just be sure that the pins are aligned correctly before applying pressure.

Installing a SIMM

The same precautions for handling and orienting individual memory chips as noted earlier in this chapter also apply to SIMMS. After all, a SIMM is a collection of memory chips soldered to a circuit board. However, removing and installing a SIMM is simple to do. The SIMM fits into a socket, and the socket has a tab at either end by which to lock the SIMM in place. Pulling outward on the tabs will release the SIMM. The SIMM may be removed by grasping it and gently rocking it back and forth out of the socket.

To install a SIMM, move the tabs outward, and press the SIMM straight down into the socket. Refer to Chapters 4 and 5 for diagrams on SIMM removal and installation.

Memory Expansion

Memory "expansion" refers loosely to the process of installing additional memory to a system. It does not refer to the type of memory that may be involved. Depending on the computer, there are three types of memory that can be added (these categories of memory are explained in Chapter 3):

- Conventional Memory
- Extended Memory
- Expanded Memory

Depending on the type and model of machine involved, additional memory may be installed:

- In available (empty) sockets on the system board
- By replacing existing RAM chips with higher capacity chips (also may require installation of an integrated circuit chip)
- By replacing the entire system board
- By installing a memory chip "package" into an adapter slot on the system

The details of all of the possible memory expansion options and devices is beyond the scope of this book. However, there are several general points that can be noted, as follows.

Conventional Memory

Today's software applications and the increasing size of DOS itself fairly well demand that any system have 640K bytes of conventional (user) memory for efficient operation. For machines having unused sockets for conventional RAM on the motherboard, expansion is a matter of purchasing and installing the "missing" chips.

If all of the motherboard RAM sockets are occupied, conventional memory can be expanded by use of an adapter card. The card contains the additional RAM chips, and is installed into one of the adapter slots on the motherboard, assuming a slot is available. Note that any amounts of memory above 640K bytes pertain to extended and/or expanded memory. PC and XT systems cannot use extended memory, but they can use expanded memory, provided it is run under DOS 3.3 or higher, and the memory conforms to the Extended Memory Specification 4.0 (EMS 4.0), or the Enhanced Extended Memory Specification (EEMS).

A common misconception among those new to PCs is that by "expanding" the memory on a personal computer, say, by installing a couple of megabytes or more of "additional" memory, that such memory will somehow function as conventional memory, thus allowing the creation of huge spreadsheets, and large, memory gobbling graphics. They are sadly disappointed to find that the additional mamory does not function as expected. The DOS MEM command (as discussed at the beginning of this chapter) still reports no more than 640K bytes of conventional memory.

Running the diagnostic software package, Checkit, on the machine being used to write this book reveals that the system has 640K bytes of conventional memory installed. It also indicates that there is 3328K bytes (3.3Mb) of extended memory on board (most of which is not being used). Finally, Checkit reports that no EMS driver is installed. An EMS driver must be installed in order to use (and for Checkit to test for the presence of) extended memory as expanded memory.

So, this PC has at least 3Mb of "additional" memory, but that in no way has converted the system to a graphics work station or similarly "powerful" machine. The point of all of this is simply that the decision to "expand" memory should consider when and how the additional memory will be used by the applications to be run, and whether the benefit justifies the cost. For older machines, the cost of expanding the memory (beyond the base of 640K bytes) may be so high as to make buying a newer, faster machine a better alternative.

The installation process is not difficult. Of course, the correct (capacity, width and speed) chips must be used. Easiest is to obtain chips having the same specifications as those RAM chips already on board. Faster chips will work, but they are more expensive and no speed benefit will be obtained from them, because the operating speed is determined by the system board (system clock speed), not the RAM chips.

Similarly, installing an adapter card to add memory is straightforward. Just be sure to obtain an adapter whose specifications match those required by your machine. For this, consult the technical manual for your machine and the technical data provided by the manufacturers of the adapters. The documentation just mentioned should include instructions on whether and how any jumpers or switches must be set on the system board or on the adapter card.

For the IBM PC, XT and AT, a "memory module kit" consisting of a set of 9 or 18 RAM chips is available for expanding conventional memory.

On PS/2 systems, memory expansion is via a small card containing RAM chips. The card is installed in a specific socket on the system board. Alternatively, memory can be added by an adapter card that fits into one of the system's expansion slots.

Expanding the conventional memory on a system board by replacing the existing chips with higher capacity chips is another matter. For example, the conventional memory on an IBM Type 1 motherboard (1983-1987, 64/265K board) may be "expanded" by replacing the existing chips with 256K chips (200 nanoseconds, or faster).

But, the "upgrade" also requires installation of a special multiplexor chip (74LS158) in the U84 socket of the system board. And, it requires installation of a jumper wire across two pins (pins 2 and 8) on the IBM proprietary chip located in the U44 socket on the system board. These changes essentially bring the XT Type 1 board up to the XT Type 2 board (640K XT, 256/640K; 1986-1987).

For those having further interest, the upgrade procedure is described fully in reference [2], page 521. The foregoing brief summary of the expansion procedure, however, illustrates the nature of the task when one decides to expand memory by other than filling empty sockets or installing an adapter card.

361

Configuring the System

Depending on the type of computer involved, the system must be informed (configured) concerning the amount of memory installed. For the IBM PC and XT, this is done by setting switches on the system board (and on the adapter cards, if any). The IBM AT and PS/2 models have no switches to set; they are configured using software. The AT, however, does have one jumper on the system board that must be set. Note that when computer documentation refers to "soft switches," it means the conditions or parameters set by using software; this has nothing to do with physical, manually operated switches.

Configuring the IBM PC and XT

The IBM PC and XT require switches (positions on switches within a "block") to be set on the system board, to configure the amount of memory installed. A switch block is illustrated in Figure 14-11. The block has 8 miniature switches, the positions of which are set to configure various options installed on the system. The PC has two such blocks (Block 1 and Block 2), while the XT has only one (Block 1).

For both the PC and XT, switch positions 3 and 4 on Block 1 are set according to how much memory is installed. The settings are shown in the right-hand columns of Figure 14-12. Additional switches must be set on Block 2 of the PC, as shown in Figure 14-13. Since the XT does not have a second block, it has no further switches to be set.

Fig. 14-11. A Typical System Board Configuration Switch

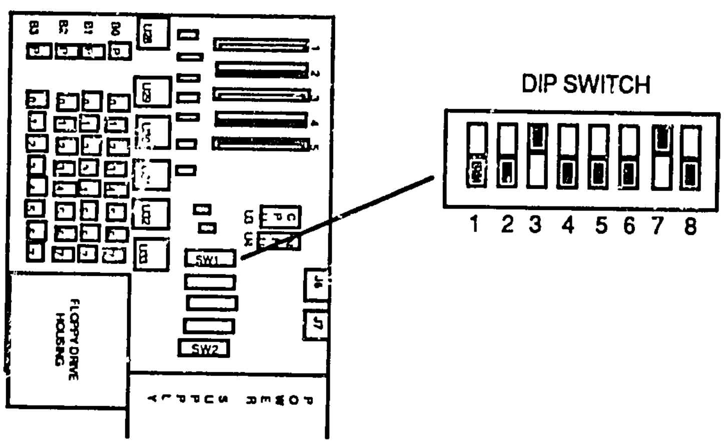


Fig. 14-12. Block 1 Switch Settings for both the PC and XT

Amount of Memory Installed		Switch Settings*	
PC	XT	SW3	SW4
16K	64K	1	1
32K	128K	0	1
48K	192K	1	0
64K or more	**265K or more**	**0**	**0**

* 0 = switch open (off/down); 1 = switch closed (on/up)

Figure 14-13 shows the Block 2 settings used only by the IBM PC (not the XT). The correct switch position settings are shown at the left side of the figure. Note that of the 8 positions, positions 6, 7 and 8 all are set to 0 (open/down), regardless of the amount of memory installed. At the right side of the figure, there are two columns. The first column pertains to the XT 64Kb system board. The second column is for XT 256Kb system board.

Below the column heading for each type of system board is shown the amount of additional memory that has been installed by means of adapter cards. That amount is in addition to the base amount indicated by the column heading. The values in parentheses show the total amount of memory installed (base on system board plus amount on adapter cards). The amount in parentheses is the value that the DOS MEM command should report when the system is correctly configured by the switch settings as shown.

Configuring the AT

Unlike the IBM PC and XT, the AT happily has no manual switches to set. But it does have one jumper (J18) that must be set when an additional bank of memory has been installed on the system board, and/or a 128Kb memory-expansion adapter has been installed. J18 is located at the right-hand corner of the system board toward the front of the machine. If the 128Kb expansion adapter and/or a bank of memory has been added to the system board, the jumper should be across pins 1 and 2. Otherwise, it should be across pins 2 and 3.

Fig. 14-13. Switch Settings on Block 2 of PC

PC Block 2 Switch Positions (1-8)*								Base Memory on System Board	
1	2	3	4	5	6	7	8	64K	265K
0	0	0	1	1				224 (284)	32 (288)
1	1	1	0	1				256 (320)	64 (320)
0	1	1	0	1	All are 0			288 (352)	96 (352)
1	0	1	0	1				320 (384)	128 (384)
0	0	1	0	1				352 (416)	160 (416)
1	1	0	0	1				384 (448)	192 (448)
0	1	0	0	1				416 (480)	224 (480)
1	0	0	0	1				448 (512)	256 (512)
0	0	0	0	1				480 (544)	238 (494)
1	1	1	1	0				512 (576)	320 (576)
0	1	1	1	0				544 (608)	352 (608)
1	0	1	1	0				576 (640)	384 (640)

* 0 = switch open (off/down); 1 = switch closed (on/up)

To configure the AT after installing memory chips and/or adapter cards, and setting the J18 jumper if necessary (as described in the preceding paragraph), insert the Diagnostic Diskette that comes with the machine and turn the power on. If the memory is not correctly configured (and we are assuming that it has not been), the POST will report an error message. When this happens, press the F1 function key to continue. From there, follow the prompts on the screen and enter the data needed. Assuming that the only change to the configuration is the memory just installed, answer "yes" to all of the questions that appear on the screen until the prompts appear for entering the amount of memory installed.

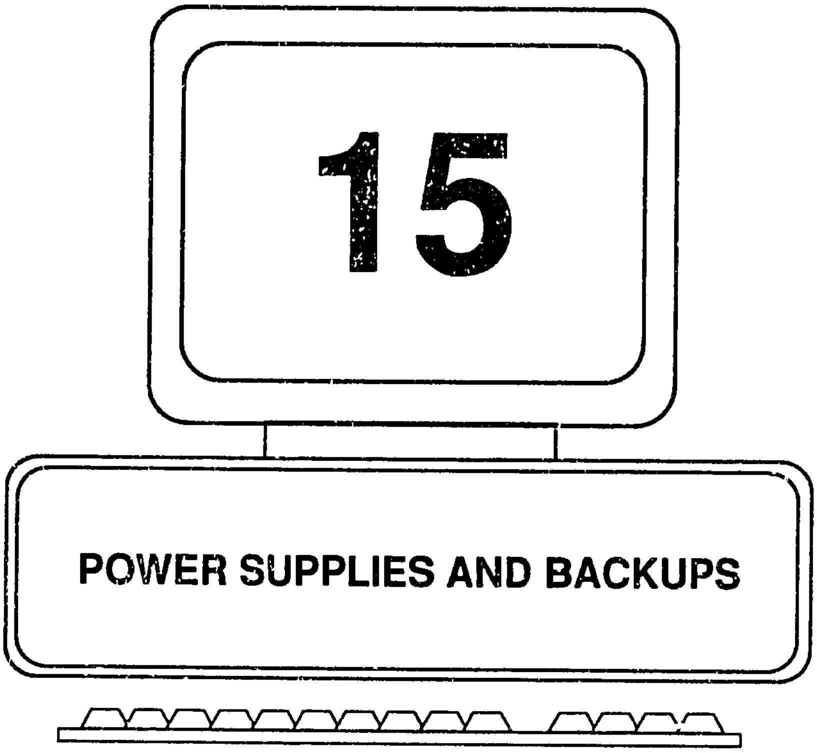
Configuring PS/2 Systems

On PS/2 systems, there are no switches to be set on the system board. Configuration is done using the configuration option selected from the main menu presented by the Reference or Starter diskette that comes with the computer. Note, however, that PS/2 Model 25 and 30 need no configuration.

The Reference/Starter diskette allows the user to simply confirm that the list of installed options is correct, set the date/time, and prepare the system for moving.

Insert the Reference/Starter diskette into the (A:) drive and turn the system on. The POST will run, the system will boot up, and then the IBM logo will appear on the screen. Press the ENTER key to continue. The main menu of the Reference/Starter program will appear on the screen. Choose option number 3, "Set Configuration." If the configuration is not correct, the prompt,"Automatically reconfigure system," will appear. Press Y to run the automatic reconfigure program. This should take care of the reconfiguration.

NOTES


15
POWER SUPPLIES AND BACKUPS

CHAPTER 15: POWER SUPPLIES AND BACKUPS

Essential to all PC operations is having the correct type and quality of electrical power. In this chapter we examine the power output (capacity) of the power supply within the various IBM personal computers. Next, we consider the nature of the main electrical power service (line power) to a home or office, and observe the various kinds of irregularities in that power which may plague the PC user. Finally, we examine the major kinds of devices available for protecting the PC from line power irregularities, and for providing power backup when the main (line) power fails.

Terminology

Since power supplies and backups are discussed in terms of their electrical parameters, such as voltage, amperage, current, power, frequency, and volt-ampere ratings, we begin with a brief review of the terminology used later in the chapter.

Voltage

Voltage is defined as the difference in electrical potential between two conductors in an electrical circuit. It is measured in volts (V) or kilovolts (kV). In electrical formulae, voltage is usually represented by the letter "E" rather than "V."

Resistance

Resistance, measured in ohms, is the extent to which anything in an electrical circuit resists the flow of electricity. In electrical formulae, resistance is represented by the letter "R."

Current

Current is the rate of flow of electricity in a circuit, measured in amperes ("amps"). In electrical formulae, current (amperage) is represented by the letter "I." However, when specifying the power rating of an electrical device, amperage is designated by the letter "A."

Frequency

Frequency is the number of times per second (cycles per second) the direction of current flow reverses itself in an electrical circuit. Cycles per second is most commonly expressed as "Hertz," abbreviated "Hz." Thus, 1 cycle per second = 1Hz; 1000 cycles per second = 1000Hz, or 1 kHz (kilo-Hertz), and so on.

Ohm's Law

The rate of flow (current) depends on both the voltage (I) and the resistance (R) in the circuit. As voltage increases, the rate of flow increases in proportion. As resistance increases, the rate of flow decreases in proportion. Amperes are calculated by dividing the voltage by the resistance, the equation for which is Ohm's Law:

I = E/R

where: I = current, in amperes

E = volts

R = resistance, in ohms

For example, a circuit having 12 volts connected to a light bulb having a resistance of 20 ohms would draw a current of 0.6 amperes:

I = 12 volts / 20 ohms = 0.6 amperes

Power

Power is the amount of electrical energy consumed by a device. Power is measured in watts; watts are calculated by multiplying amperes, which is "current" times voltage.

P(watts) = (I) amperes x E (volts)

For example, given a 12 volt circuit connected to a device that draws 0.6 amperes of current, the wattage consumed would be 7.2 watts:

P = 0.6 amperes x 12 volts = 7.2 watts

Since power is defined as the arithmetic product of volts (V) and amperes (A), the power rating of a device may be expressed either in watts (W), or in volt-amperes (VA). In the notation of physics and electrical calculations, a hyphen or dash between two units of measure indicates that the units have been multiplied together. Thus, volt-amperes means "volts x amperes." The rating for devices that consume large amounts of power may be expressed as kilovolt-amperes (kVA). As an example, the following ratings are equivalent:

1000 Watts (1000W)

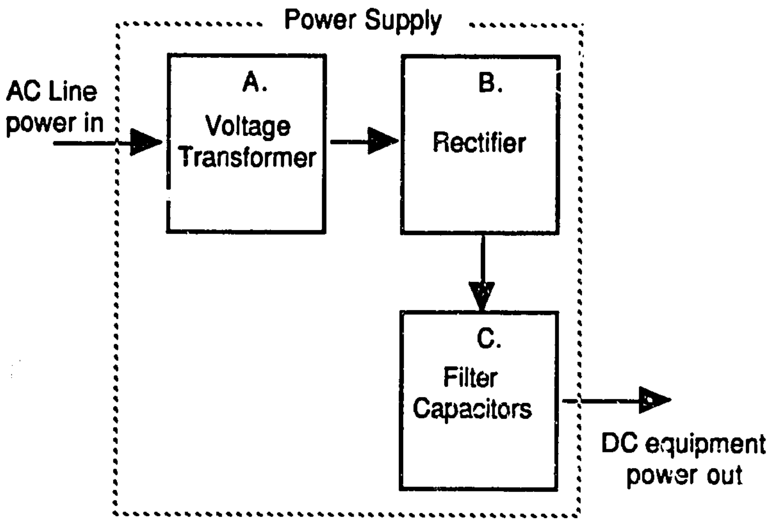
1000 volt-amperes (1000 VA)

1 kilovolt-ampere (1kVA)

The Personal Computer Power Supply

Within each PC is a power supply. The purpose of the power supply is to convert incoming, main AC power into direct current (DC) power for use by the computer. Figure 15-1 illustrates the basic elements of a PC power supply.

Fig. 15-1. Basic Elements of a PC Power Supply



A. Transformer increases or decreases voltage, according to power supply design.
B. Rectifier converts cyclical form of current to one phase (positive or negative).
C. Capacitors "smooth out" current, eliminating residual variations and spikes. Capacitors hold a charge, like a battery. That is what makes them dangerous.

The supply does not create (i.e., does not generate) power, neither does it contain batteries to serve in a backup capacity. Of course, this does not apply to certain portable or lap-top computers which operate directly from internal, rechargeable batteries. The latter type of computers require main AC power to recharge the batteries; and alternatively, they may use the main AC power when it is available, just as a "regular" PC does.

The PC power supply provides current to operate the cooling fan, disk drive motors, disk drive and keyboard display lamps, and all of the electronic components within the system, including all of the adapter boards and circuitry installed. Note that certain PCs may have a female, AC receptacle on the back panel. If so, it will most likely have a configuration different from the normal AC connectors. This outlet is intended for use by a specific type of monitor (usually an IBM monochrome monitor, in particular the IBM PC and the IBM PS/2 Model 25).

The outlet should not be used for any other purpose, and no attempt should be made to defeat the shape configuration of the receptacle, such as by using any type of mating connector or adapter. Except for the specific type of monitors just noted, each display monitor should be plugged into its own, separate wall outlet. In no event should a 120v monitor be plugged into a 230v outlet! This will destroy the monitor, and it could be dangerous for anyone standing nearby.

Each PC power supply has a rated output capacity, specified in watts. The primary concern is that the total power drawn by all of the components supported by the power supply does not exceed the capacity of the power supply. Early models of personal computers, notably the IBM PC, were provided with weak power supplies (i.e., only 63.5 watts of output power). Installing adapter boards and other optional components thus required upgrading (replacing) the PC's power supply, which normally included a more powerful fan for cooling the system.

Figure 15-2 lists the power supply output (watts) for various IBM personal computers. Note that the figure shows the wattage provided as output from the power supply, not the power drawn from the main AC line by the power supply itself. The figure also indicates whether and how the AC input power may be selected for the machine.

It is interesting to note from the figure that as PCs have become more powerful in a functional sense (more capabilities, more memory, larger and faster disk drives), their consumption of electrical power has grown almost in proportion. From the humble IBM PC with its 63.5 watt power supply, we now see the PS/2 Model 60 and 80 power supplies putting out 225 watts, nearly a 400 percent increase over the earliest PCs.

371

Fig. 15-2. Power Supply Output for Various IBM Computers

IBM Computer	Power Supply Output, Watts	AC Input Selection (120v/230v)
PC	63.5	None:fixed (120v)
XT	130	None:fixed (120v)
XT-286	157	Automatic
AT	192	Switch
25 (color)	190	Manual
25-286 (color)	124.5	Manual
30	70	Automatic
30-286	90	Manual
50, 50Z	94	Automatic
55SX	90	Manual
60-041	207	Automatic
60-071	225	Automatic
65SX	250	Automatic
70	132	Automatic
P70	85	Automatic
80-041	207	Automatic
80-071	225	Automatic

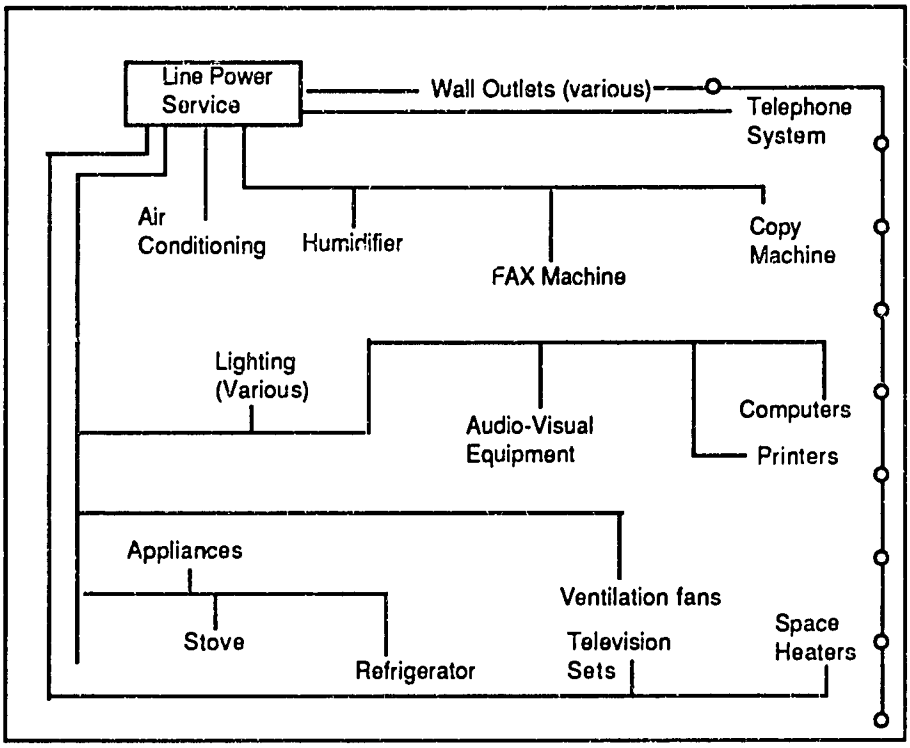
Power Main (Line Power)

A power "main" is the electrical power provided to your home or office building by the power company or power authority. It is the electrical power available at the wall outlet. This power is described in terms of its voltage and its frequency. Also, any given power line can provide only a certain amount of current (amperage). For example, the power line coming into an office may provide 200 amps of 120 volt, 60Hz power to the main electrical box ("breaker box," "fuse box" or "junction box"). That power is then distributed over wires to the various rooms and wall outlets in the building. Each such wire will have an assigned amount of amperage, its share of the total of 200 amps coming into the building. Assuming, for example, that there were 10 such lines, and each provided 20 amps of power, then the total number of devices (coffee machines, computers, etc.) connected to any one line should not demand more than a total of 20 amps from that line.

Figure 15-3 illustrates how incoming power might be distributed within an office. Note how the various electrical devices cumulatively consume power from any line that they share.

372

Fig. 15-3. Example of Power Distribution within an Office



In a typical office, many appliances may be drawing a wide range of power. Excessive pwer demands on a line shared by computters can cause problems with the computers.

Perhaps because the voltage provided by the main is not exactly constant, one encounters different voltage numbers that, in effect, mean the same thing. Basically, what the power company provides is electricity within a specified range of voltage. Under normal conditions, one may expect the voltage to be within that range. In Western countries, the voltage range is typically 110 - 125 volts. This is referred to as the "120 volt range." This power has a frequency of 60Hz.

In contrast, many Middle-Eastern countries for example, have mains operating in the range of voltage from 210 to 230 volts. This is referred to as the "220 volt range." This power has a frequency of 50Hz. As noted below, it is essential that the main power provided to electrical devices be within the operating range required by the devices. An improper match can have very serious consequences, including electrical hazard, fire/smoke, explosion, and destruction of the electrical device.

Computer System Power Requirements

Electrical devices (television sets, computers, display monitors, printers) are designed to operate on main power within a specified range of voltage and frequency. Further, each device requires a specified amount of the correct power (current, amperage) for it to operate properly. It is important to remember that certain devices, such as a personal computer, may have internal components and circuitry that will accommodate either 120v, 60Hz power; or 220v, 50Hz power. The computer may have a selector switch on it that allows a setting to one type of power or the other. Or, the computer may have no such circuitry, and thus would require an external power converter in order to operate.

Furthermore, peripheral devices attached to a PC, such as the monitor and printer, each has its own main power requirements. Just because the PC itself may have a power selector switch on it that will satisfy its own main power requirements, this has no bearing whatever on the power supplied to the monitor, printer or other peripheral device. All devices that have a power cord designed to be plugged into a wall outlet must be provided the power voltage and frequency they require. Otherwise, the device either will not operate or it could be destroyed altogether.

As mentioned earlier, each device consumes a certain amount of power. All devices connected to a common source of power (e.g., a wall outlet, or a multi-plug extension cord, or a particular power line serving other devices in the office or building) compete for the power available from that source. The power demands of these devices are additive (cumulative). If too many devices are connected to a common source of power, the demand will be excessive, resulting in inadequate or degraded power being supplied to the individual devices -- each device will not receive the amperage and voltage it needs to operate properly and safely.

Irregularities in the Power Main

At one time or another, the power supplied to a home, building or office will deviate significantly from its normal specifications. It may be interrupted for just a fraction of a second, or it may be off for hours or days at a time. When the power is on, it may exhibit sudden and severe increases in voltage (called "spikes") lasting only a small fraction of a second, or increases of longer duration, called "surges." Or, the power may operate for extended periods at voltage below the normal range (undervoltage), or above the normal range (overvoltage).

Some of these irregularities may be due to environmental conditions, such as lightning storms which can send extremely high voltages through the power system and through the devices connected to the system. Irregularities can be caused also by the malfunctions in the generation and transmission of the main power.

Finally, problems may be caused locally, by the operation of equipment in the building. Any and all of these irregularities can have serious consequences for computer operations: devices may be damaged, and data may be lost or corrupted. Depending on the likelihood of power irregularities in a given geographic location or work setting, several kinds of devices are available to minimize the undesirable effects of faulty power. These include surge protectors, power line conditioners, standby power supplies, and uninterruptable power supplies.

Power Conditioning and Backup Devices

Power conditioning is concerned mainly with ensuring that the line power coming in to a PC and its peripheral devices is within its normal operating range levels, and that it is not contaminated by either sudden or prolonged variations from those levels. Power backup devices are devices that provide an independent source of electrical power in the event the main (line) power fails.

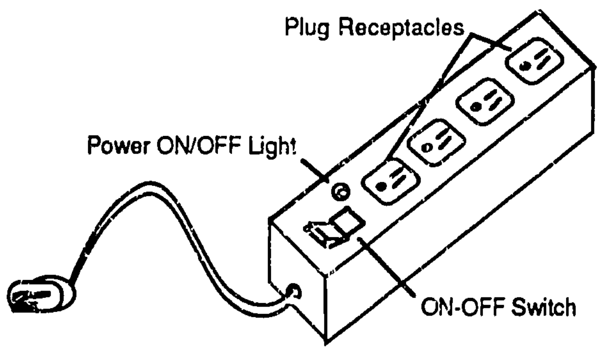
Surge Protectors

As its name suggests, a surge protector is an electronic device designed to prevent excessively high levels of voltage from passing through it. In essence, these devices contain a voltage sensitive component that acts as a switch. When the incoming voltage exceeds a prescribed level, the switch closes and routes ("shunts") the excess power to the ground side of the circuit, thus protecting the equipment connected to the surge protector.

As shown in Figure 15-4, surge protectors are relatively simple devices. One end plugs into the wall outlet. The other side of the device has one or more sockets into which the protected equipment is plugged. Typically, the device has an on/off power switch on it, such that all devices connected to it can be turned on or off at the same time. Often, one will see a surge protector serving the PC, printer and display monitor, as a kind of multi-socket extension cord. The use of these types of surge protectors and the practice of plugging multiple components into them, however, is open to serious question.

Many of these surge protectors are designed to "protect" on just one occasion. Once they are exposed to a voltage sufficiently high to trigger the shunt mechanism, the shunt will work on that occasion but it will no longer work in the future. The device, however, will continue to pass current through it, and the user will be unaware that the protector no longer functions as intended.

Fig. 15-4. A Typical Surge Protector



A second problem is in using the protector as a multi-component, on-off switch. When an electrical device is first turned on, it is "cold" relative to its normal operating temperature. At lower temperatures (e.g., even at room temperature),

electrical devices have lower electrical resistance. So, when the power is turned on, the device draws much more current that it normally would. It causes a "surge" in power demand. Now, if several devices all are connected to the same power line (e.g., a surge protector, multiple outlet) and all are turned on at the same time, they all contribute cumulatively to the initial power surge. This is an unhealthy situation, and it is best avoided by turning on the devices one at a time, with the PC being turned on last.

Line Conditioners

A power line conditioner is a device designed to regulate the flow and quality of electrical current, keeping the characteristics (parameters) of the power within acceptable limits. The device, in essence, acts as a filter and a compensator, to stabilize the voltage and to remove spikes and surges. As these functions are usually included in most high quality backup power supplies as noted below, further discussion of line conditioners will not be entertained here. Suffice it to say that a line conditioner, in and of itself, is not a power supply.

Step-Down Transformers

A transformer is an electrical device that receives one level of voltage as its input, and produces a lower level of voltage as its output. Thus, as its name suggests, the transformer "steps the voltage down." There are, of course, devices that do just the reverse, and as one might suspect, these are called "step-up" transformers.

One of the components of a power supply is the transformer. Its job is to reduce (step down) the incoming 120 or 230 volt AC power to 12 volt and 5 volt AC power. The transformer by itself, however, does not alter the frequency (Hz) of the power. A transformer merely changes the voltage. It does not convert AC power into DC power. The AC to DC conversion is performed by other circuitry in the power supply.

Of what practical use, then, is a step-down transformer? The answer is that such a device may be used when the line voltage needs to be reduced in order to supply a device which requires a lower voltage, but which either uses the original line frequency or is relatively indifferent to that frequency. For example, a step down transformer may be used to convert 230 volt, 50 Hz current into 120 volt, 50 Hz current. If the device that requires the 120 volt AC current can accept a frequency range of, say, from 40 to 70 Hz without incurring damage or a malfunction, then one can "get away with" not providing the optimal power frequency for that device, e.g., 60Hz.

This procedure may succeed, for example, where the 120 volt device in question does not use the incoming AC power directly to operate frequency-sensitive components. Perhaps the incoming 120v, 50Hz power is used only to charge up some internal batteries, and the batteries in turn provide all of the internal power needed by the device. In such instance, the battery charging circuit and the batteries themselves may be acting as a kind of "frequency filter," and so the device functions.

In such cases, however, a mismatch between the power supplied and the power required by the device may produce negative effects over prolonged use. Among the possible undesirable effects are excessive heat generation and electrical noise (unwanted electrical energy or signals) within the device being so operated. If at all possible, a proper power supply should be used. As an expedient, however, a step down transformer at least reduces the potentially disastrous consequences of supplying a device with an excessively high voltage.

Standby (Backup) Power Supplies

There is no small amount of confusion surrounding the issue of so-called "backup" power supplies. In the generic sense, a backup supply is any alternate source of power that can be used when the primary source of power fails. Since the main power is alternating current (AC), an alternative source of AC power is needed. The alternative power is provided by batteries, which are a source of direct current (DC) power. The DC power from the batteries is converted to AC by means of circuitry within the device, and the AC power thus produced is provided to run the computer and its peripheral devices. An auditory alarm is also provided to alert the user that the backup system has sensed a failure of the main AC power. At this point, a major distinction occurs between backup power supplies, and that is whether or not they are "switched."

Both types of backup system, switched and non-switched, contain batteries, and when the main AC power is on, the AC power is used to keep the batteries charged. But, the devices operate in a significantly different manner. With a switched supply, an electronic switch is activated automatically when the main AC power fails. This ties the batteries into the circuit, and the device begins to provide AC "backup" power through the procedure described in the preceding paragraph. The basic arrangement of a switched backup supply is illustrated in Figure 15-5.

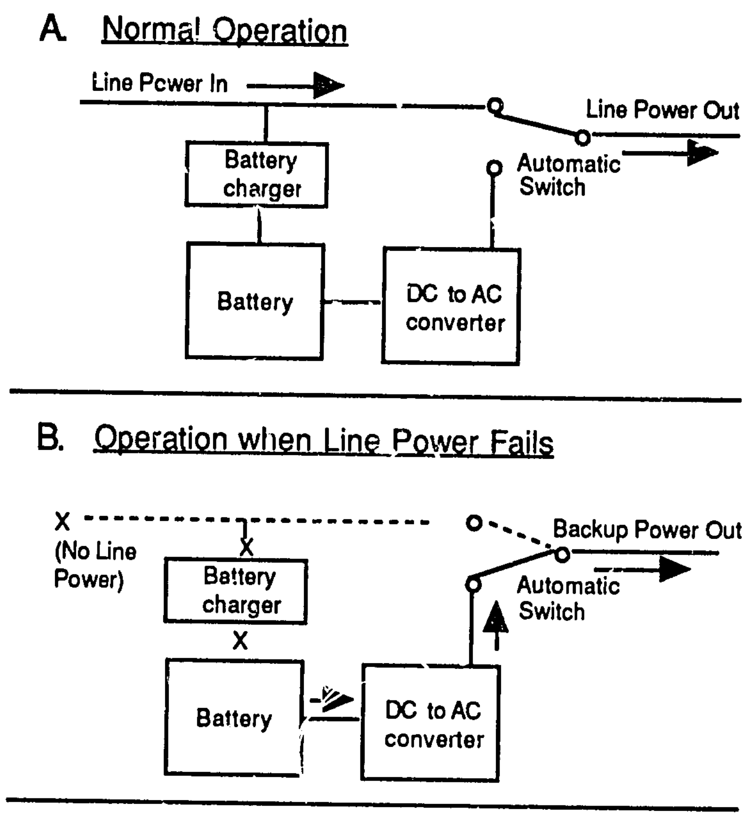
Switched power supplies, unfortunately, are often advertised as being "uninterruptable" power supplies. They are not uninterruptable. Any backup supply that must be switched on (internally or otherwise) in the event of a main AC power failure is, by definition, an interruptable supply. It does not matter how fast the switching occurs.

Any backup power supply that specifies a switching speed or switching time interval is not an uninteruptable power supply. Furthermore, the internal power supply of the PC is designed to shut down immediately in the event of a line (main AC) power failure. The internal supply, therefore, may shut down before the switched backup system is able to react. So with or without the switched supply being present, the power would be interrupted, and an orderly shut down of the computer would be prevented.

For regular, non-digital household appliances such as lamps, vacuum cleaners and refrigerators, a switched supply may be perfectly adequate. For a computer, however, it may be far less than sufficient. It depends in part on the activity being performed by the computer when the main power fails (such as writing an important file to disk), and this is impossible to predict.

In brief, a power supply as described above does not use its batteries until there is a failure of the main AC power. This is in contrast to an uninterruptable supply, as described below.

Fig. 15-5. Basic Design of a Switched Backup Power Supply



Panel A shows line power going straight through to ouput of supply. Line power also keeps battery charged.

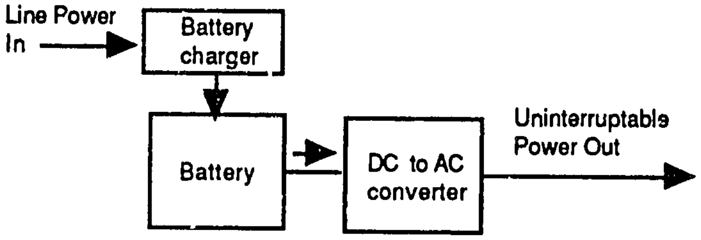
Panel B shows how, when line power fails, power is supplied by a battery. But the connection must first be switched from the line (normal power) the ouput of the DC to AC converter.

Uninterruptable Power Supplies (UPS)

A true "uninterruptable power supply" (UPS) works in the following way. Main AC power is used to keep the batteries charged. The batteries are continuously in service, providing DC power which is continuously converted to AC power and fed to the computer and other devices being supported. The basic wiring of an uninterruptable power supply is illustrated in Figure 15-6. As may be seen, there is no switching on or off of the batteries. When the main AC power fails, the batteries continue to provide power until they run down, i.e., until they are discharged. The amount of operating time provided by the uninteruptable power supply after the main AC power fails depends on the number and capacity of the batteries present, and on the amount of current drawn by the devices the power supply is supporting.

Smaller, lower cost supplies may provide just a few minutes of backup power. The purpose of these systems is to allow the user enough time at least to shut down the computer in an orderly manner. Larger, more expensive systems may consist of many banks of batteries, thus providing more operating time. In turn, these batteries may be supported by fuel-operated generators to keep them charged when main AC power is not available. The generators may be switched on manually or automatically when the main AC power is interrupted.

Fig. 15-6. Basic Design of an Uninterruptable Power Supply



With an uninterruptable supply, there is no switching when the line power fails. The system continues to run off of the battery and converter as shown here, until the battery becomes discharged.

A good quality, uninterruptable power supply will also act as a line conditioner. It will prevent surges, spikes and other power irregularities from reaching the computer. For this reason, when a UPS is being used, it is not necessary to use separate surge protectors or line conditioners.

When selecting a backup power supply, the total power needed by the computer and its peripherals must be considered. In so doing, also evaluate the possibility that the backup power supply may have to support some lighting by which to work (i.e., a desk lamp), in case the power failure occurs at night. The procedure for calculating the needed output capacity of the backup power supply is as follows.

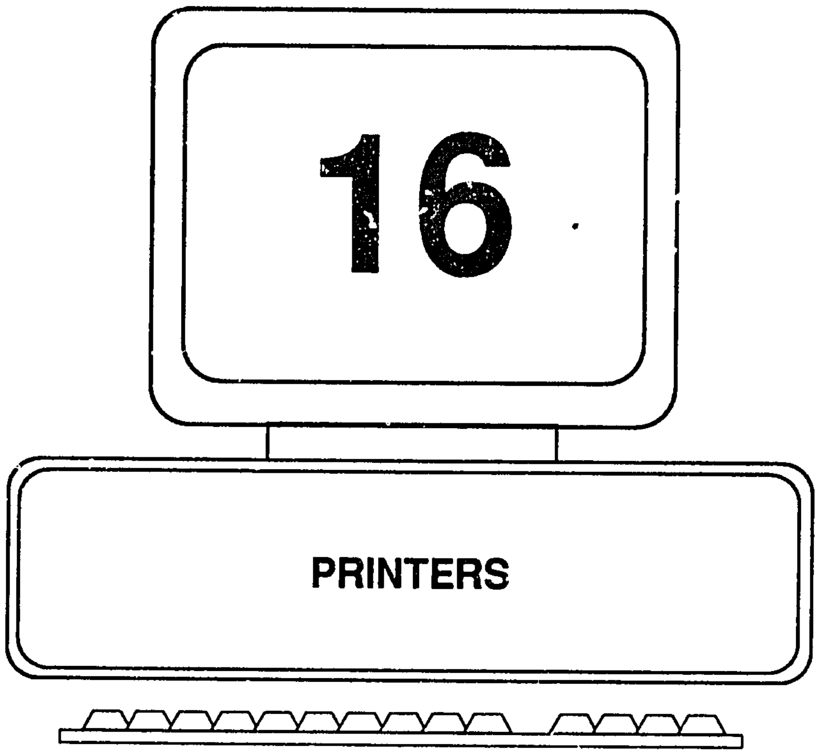
For each device (including the computer), find the VA rating by multiplying the current (amperes) drawn by the device, times the voltage at which the device operates. Do this for all of the devices to be supported by the backup power supply, and add up the VA values to find the total power needed. Figure 15-7 presents a simple example of the calculations.

In this example, we see that a minimum of 4200 volt-amperes (VA), or 4.2kVA, is needed to support the equipment listed in the table. The next question is how long the backup supply should last. The power provided by a battery or a battery-based power supply is expressed as the amperes provided (drawn) at the specified voltage level, multiplied by the length of time the supply will last. For example, a 12 volt battery may be described as being a 100 ampere-hour battery. This means that the battery will provide a current of 100 amperes for a duration of one hour.

Fig. 15-7. Calculating Backup Power Supply Capacity Needed

Device	Amps	Volts	VA	No. Devices	Total VA
Computer	1.4	120	168	5	840
Display	0.8	120	96	5	480
Laser Printer	6.0	120	720	3	2160
Desk Lamp	1.2	120	144	5	720
Total					4200

In specifying the requirements for a backup supply, indicate both the kVA output requirement and the operating time in minutes or hours. In our example, we may elect to purchase somewhat more than the minimum capacity needed, and further to indicate that the supply should run the equipment for at least two hours when a line power failure occurs. Thus, we might specify a two-hour, 5.0 kVA supply. Note that a more economical approach would be to eliminate support for the printers, and the desk lamps (perhaps by having some flashlights handy). This would reduce the requirement to only about 1.4 kVA, and the cost of the supply would be proportionately lower.


16
PRINTERS

CHAPTER 16: PRINTERS

A thorough treatment of the subject of printers would fill many volumes. Here we can but touch the surface, and cover a few points that may be helpful to those totally unfamiliar with the world of printers. We begin by looking briefly at how the most commonly encountered printers work, and how they can be quickly tested. Next we review the basics of parallel and serial communications, and the pin assignments for cables connecting the computer to the printer. This is followed by a brief explanation of how to set the communications parameters when using a serial printer.

Finally, we look at the mysterious world of printer control codes and some of the perplexing commands that make up the "escape sequences" by which printers are controlled. We conclude with a short dBase program that illustrates how printer setup codes for the HP Laserjet II can be made easier to use by placing them in a menu.

How a Laser Printer Works

A laser printer works very much like a copying machine does. A piece of clean, white paper goes in one side, and it comes out the other side with text or graphics printed on it. The characters and lines printed on the paper consist of microscopic particles of a black powder ("toner") that have been melted (fused) and then solidified to cling to the paper. The particles are grouped into tiny dots, and the dots are arranged to form the images on the paper. The smaller the dots are, and the more of them that can be packed into one inch of space, (horizontally and vertically) the higher is the "resolution" of the printer – the sharpness and clarity of its images.

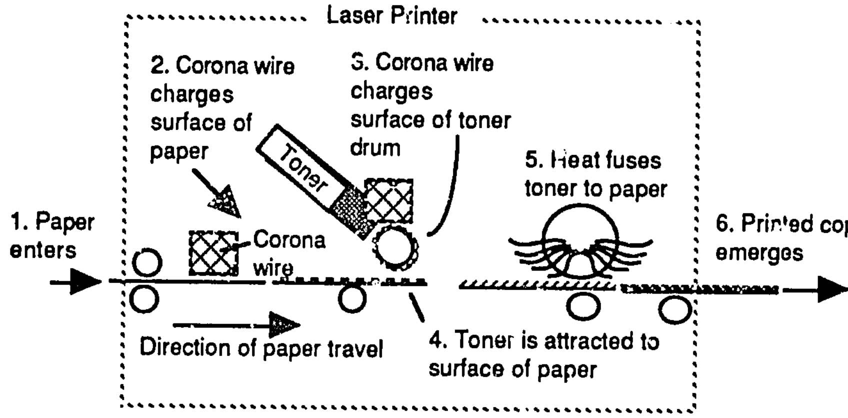
Laser printers commonly have a resolution of 300 Dots Per Inch (DPI). The diameter of each dot is therefore 1/300th of an inch. In a 1 inch x 1 inch square area on the paper, a 300 DPI printer could fill the space with as many as 300 x 300, or 90,000 dots. More expensive printers can have a resolution of 1200 DPI or greater. On the other hand, FAX machines typically operate at about 120 DPI.

As the paper passes through the printer, several things happen, as shown in Figure 16-1. First, a "print drum" (a cylindrical bar) has a pattern of electrostatic charges placed onto its surface. The electrical charges are produced by wires called "coronas." The pattern of electrical charges on the drum corresponds to the image (text, etc.) to be printed. The charges attract the microscopic particles of toner, which cling to the drum, forming in black powder the pattern to be printed.

The paper receives an electrostatic charge opposite in polarity to that on the drum. Recall from Chapter 6 that magnetic charges opposite in polarity attract each other.

As the paper passes close to the drum, the carbon particles are attracted to the paper. In essence, the particles "jump" from the drum onto the paper, keeping the pattern that was formed on the drum. While the particles are clinging to the paper, they are exposed to heat from a "fusing roller." The heat melts the particles, causing them to adhere to the paper. The paper is then transported on its way, and out of the printer.

Fig. 16-1. How a Laser Printer Prints



The toner is supplied by a cartridge that slides into the printer. The cartridge also contains the print drum, and the main corona wire. The rest of the elements needed for printing are part of the printer itself.

The pattern created on the print drum and later transferred to the paper is created in response to digital signals received from the computer. The "pattern" originates when a document is typed, say in a word processing application. The document contains not only the characters to be printed, but other information that tells the printer when to begin a new line; whether to print in bold, underline, or italics; what the margins should be; whether the page should be printed across its narrow side (e.g., across the 8 1/2 inch dimension, referred to as the "portrait" orientation) or across its wide side (e.g., across its 11 inch dimension, referred to as the "landscape" orientation). These and many other instructions are sent to the printer from the computer. Alternatively, many such specifications can be set on the printer itself.

381

Installing a Toner Cartridge

If a cartridge is already installed and beginning to run out of toner, a "toner low" message or icon will appear on the printer front panel. When this occurs, open the printer and slide the cartridge out. Hold the cartridge by its long sides, one hand on either side, and gently shake the cartridge by alternately lifting one side up while lowering the other. In other words, rock the cartridge back and forth. Tilting it to an angle of 45 degrees each time is sufficient. Vigorous shaking should be avoided.

Repeat the actions several times in quick succession. This will distribute the remaining toner and permit the maximum useful life of the cartridge. Eventually, however, light streaks will begin to appear on the printouts, and the cartridge will have to be replaced. Under normal use in printing text documents, a cartridge will last for about 3500-4000 copies. Printing graphics containing shaded or black areas will consume toner at a much faster rate.

To install a new cartridge:

1. Turn the printer off (safety precaution).

2. Open the printer cover (be careful -- certain areas are very hot!) and remove the existing expended cartridge, if present.

3. Remove the new cartridge from its pouch; DO NOT yet pull out the strip of plastic that holds the toner in place; there should also be a long, slender object in the package; this is the cleaning pad for the fuser roller, to be installed shortly.

4. Distribute the toner by rocking the cartridge gently back and forth (as described earlier concerning a low toner cartridge).

5. Slide the cartridge into the top of the printer, between the guides on the underside of the cover.

6. At the left side of the cartridge toward the back of the printer, find the plastic tab; break the tab free; and pull the plastic strip sideways out of the cartridge. Try not to break the tab off. If it breaks, grasp the plastic and carefully pull it out.

7. Unwrap the Fuser Roller Cleaning Pad mentioned in Step 3.

8. Lift the Fusing Roller Assembly Cover; if an old pad is present, remove it and discard it.

9. Use the fabric end of the Fuser Roller Cleaning Pad to clean the slot where the roller will be inserted; then, remove the fabric from the end of the pad and discard the fabric.

10. Insert the Fusing Roller Cleaning Pad into its slot.

11. Find the small brush stored inside the printer; it is usually on the left side of the space where the cartridge rests after the printer cover is closed.

12. Use the small brush to geently clean the corona wires; a criss-cross filament may be shielding the corona wire; if so, use a foam-tipped swab to work between the filament threads and remove any accumulated toner or dust from the corona wire beneath them.

13. Vacuum or use canned air (see Chapter 9) to remove dust, paper chaff, and toner particles from other surfaces inside the printer.

14. Close the printer cover.

15. Turn the printer back on.

How an Impact Printer Works

An impact printer works very much like an old fashioned typewriter. It forms an impression (image or pattern) on paper by striking a "ribbon" against the surface of the paper. The ribbon contains material such as an ink compound that transfers to the paper. The ribbon is struck by the printer head and the force of the impact presses the ribbon against the paper, transferring the ink to the paper.

There are two main ways by which impact printers form characters on the paper. The character is either pre-formed on the print head, like the raised or embossed characters on an expensive wedding invitation; or, it is formed by a pattern of individual dots created by a rectangular array (matrix) of "pins" on the face of the print head.

Printers that use pre-formed characters are called "daisy wheel" printers, so-named because the round metal disk that contains the pre-formed characters resembles a daisy. The disk rotates to position each character to be printed, and an electromagnet drives the daisy wheel against the ribbon, which in turn strikes the paper behind it.

Printers that use a matrix of pins to strike the ribbon and thus to create ("draw") characters by patterns of dots are called "dot matrix printers." Dot matrix printers have been the mainstay of the PC industry for many years. But, like their daisy wheel cousins, they are noisy. Both types of impact printers are increasingly being replaced in favor of the silent, high resolution laser printers, when they can be afforded.

Tips on Using Impact Printers

- Print heads run very hot -- do not obstruct ventilation and keep hands away from head until head cools.
- Clean the heads (after they are cool) with cleaning solution or isopropyl alcohol; use foam tipped, lint free swab.
- Before changing a ribbon, observe the path of the ribbon around the rollers and guides, then diagram the path; use the diagram to install the new ribbon.
- Never turn the platen (i.e., do not twist the end of the roller to feed paper through) while the printer is turned on (the platen is operated by a stepper motor that is engaged to the platen when the printer is powered on). Turning the platen will damage the motor.
- Vacuum printers at least weekly, to remove paper particles (chaff).
- Be sure the correct types of cable are used and that the cables are wired properly (See Figs 16-3 and 16-4).
- Be sure printer cable connectors are securely screwed or clipped in place; a loose connection can produce printing problems that might be mistaken for software problems.
- Run the printer self-test and save a copy for future reference.

381

Printer Tests

Printer tests may be found in the form of software programs, or as tests built into the printer itself (printer-resident tests).

Software-Based Tests

Printer tests are included on the Diagnostic/Reference diskettes provided with every IBM personal computer. To use the programs, insert the diskette into the (A:) drive and turn on the power to the computer. Then follow the menu prompts to run the printer tests.

Third party software (such as Checkit; see Chapter 11)) also provides a range of software-based test for printers. The tests are run simply by selecting them from the program menu.

Printer-Resident Tests

Impact Printer Tests. For most impact type printers, a printer self-test may be run by doing the following (or a minor variation thereof – consult the printer manual). The printer does not have to be connected to the computer to run a self-test.

1. Turn off the printer power switch.
2. Press and hold down the LINE FEED button on the printer control panel, and meanwhile turn the power back on.
3. When the printing begins, release the LINE FEED key.
4. To interrupt the test, press the ON-LINE or SELECT key (whichever way it is labeled); press again to resume the test.
5. To cancel the test, turn the power off.

Laser Printer Tests. Laser printers usually have several built-in tests, to cover the wide range of print variables they provide. As with the many other electronic devices in the world today, the number of functions to be used or tested exceeds the number of "buttons" on the device's operating panel. So, it has become necessary to use other approaches, such as pressing a series of buttons, pressing multiple buttons at a time, and even having to hold a button down for a prescribed time interval before the desired function will be activated. For these reasons, trial and error is an inefficient way to decode the combination to the underlying tests and features. The only sure method is to read the printer manual.

As examples of laser printer built-in tests, here are two quick tests found on HP Laserjet II printers. The printer need not be connected to the computer. If is connected, the computer power may be on or off – it makes no difference to the tests.

Print Fonts Test. This test will print out the "permanent" fonts in the printer, as well as any fonts in cartridges installed in the left and/or right slots at the lower front of the printer.

To run the test:

1. Press the ON-LINE button – turn the ON-LINE light off.
2. Press the PRINT FONTS/TEST button briefly. The "FONTS PRINTOUT" message will appear in the LED window.
3. To terminate the test, press the ON-LINE key (several more pages may be printed out, however).

Self-Test (Built-In Controller). The self-test tests the printer's built-in controller. It also reports the following:

- Number of copies that the printer has produced (good to know for preventative maintenance purposes)
- Program ROM date
- Internal Font ROM date
- Whether the AUTO Continue is set ON or OFF
- The amount of installed memory (printer buffer)
- The Symbol Set being used by the printer
- The items currently set on the printer front panel menu
- Whether the printer interface (I/O) is parallel or serial

To run the test:

1. Press the ON-LINE button — turn the ON-LINE light off.

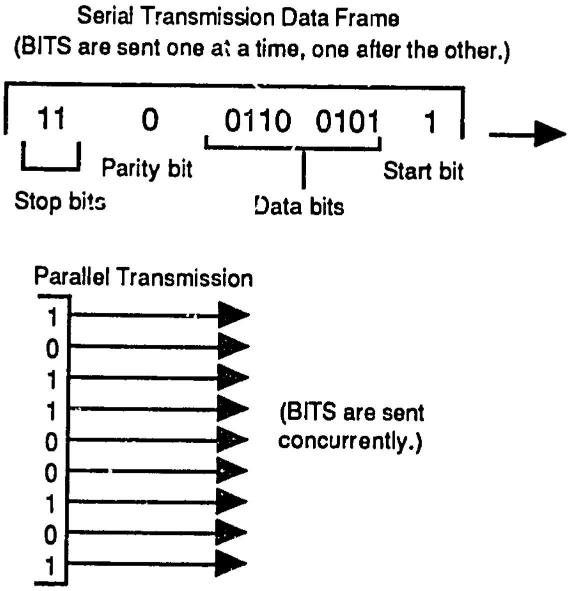
2. Press the PRINT FONTS/TEST button and hold it until the "SELF-TEST" message appears in the LED window.

3. The test will run, and a single page of results will be printed out.

Printer Cables

Some printers use a parallel interface, others use a serial interface, and some (like the HP LaserJet II's and III's) can use either one. Figure 16-2 illustrates the basic difference between serial and parallel transmission. In a serial transmission, only one data bit at a time is transmitted; in a parallel transmission, 8 data bits are sent simultaneously. Obviously, parallel transmission is much faster.

Fig. 16-2. Serial vs Parallel Data Transmission



Parallel transmission, however, is more vulnerable to electronic interference and noise than is serial transmission. The likelihood of such interference is greatly increased as the length of the communication cable increases. (The cable acts as an antenna and is thus able to pick up unwanted electrical signals.) For this reason, serial transmission is mainly used for communications transmissions. However, since a printer normally resides only a short distance from the computer, it has a short cable. So, parallel communications can be used with little or no worry about stray electrical signals interfering with the transmission.

Figure 16-3 shows the parallel cable pin assignments for an HP Laserjet Series II printer. It applies to connecting the printer with the IBM PC; PC/XT, AT and PS/2 Models 30/50/60/80.

Fig. 16-3. Parallel Printer Cable Pin Assignments for HP Laserjet Series II (re: IBM PC; PC/XT, AT and PC/2 Models 30/50/60/80)

Signal Name	Printer Pin No.	IBM Port Pin No.
nSTROBE	1	1
Data 1	2	2
Data 2	3	3
Data 3	4	4
Data 4	5	5
Data 5	6	6
Data 6	7	7
Data 7	8	8
Data 8	9	9
nACKNLG	10	10
BUSY	11	11
CALL (PE)	12	12
SELECT	13	13
LOGIC GND	14	14
nFAULT	32	15
0 VDC (GND)	19 thru 30	18 thru 25

Figure 16-4 shows the pin assignment for a serial cable between the computer and printer for the above models of computers. Figure 16-5 shows the serial pins assignments for the IBM AT.

Fig. 16-4. Serial Printer Cable Pin Assignments for HP Laserjet Series II (re: IBM PC; PC/XT, AT and PC/2 Models 30/50/60/80)

Signal Name	HP Laserjet (Male Connector)	Computer (Female Connector)	Signal Name
Chassis GRND	1 ———>	1	Chassis GRND
RD	3 ———>	2	TD
TD	2 ———>	3	RD
Sig Gnd	7 ———>	7	Sig Gnd
		5	Clear to send
DTR	20 ———>	{ (both)	
		6	Data Set Ready

Fig. 16-5. Serial Cable Pin Assignments for HP Laserjet Series II (IBM AT)

Signal Name	HP Laserjet (Male Connector)	Computer (Female Connector)	Signal Name
TD	2 ———>	2	RD
RD	3 ———>	3	TD
Sig Gnd	7 ———>	5	Sig Gnd
		6	Data Set Ready
DTR	20 ———>	{ (both)	
		8	Clear to send

Setting the Serial Communication Parameters

A serial printer uses a communications port (COM port) on the computer, usually an RS-232C port. To use a serial printer, the communications parameters must be set on the computer. There are four parameters:

- Speed/Baud Rate (1200, 2400, 4800, 9600)
- Parity (Even, Odd or None)
- Number of Data Bits (7 or 8)
- Number of Stop Bits (normally 1 or 2)

The values to be set for these parameters are specified in the printer's manual. The values are set by placing them in a DOS MODE command, which is stored in the computer's AUTOEXEC.BAT file. Then, device LPT1 is set to COM1, so that the COM output is directed to the printer (LPT).

The sequence of values for the MODE command is:

Device: speed, parity, data bits, stop bits, P

Each value is separated by a comma. If no value is entered, a comma is used for the "missing" parameter anyway, to keep the positional sequence correct. The sequence of characters ends with a "P," to denote a printer. The following example illustrates the procedure.

Assume the printer manual calls for the following parameter values:

- Speed/Baud Rate = 4800
- Parity = Even (coded as "E")
- Number of Data Bits = 7
- Number of Stop Bits = 1

Using a the DOS text editor (EDLIN or EDIT), write the following command:

```
MODE COM1: 4800,E,7,1,P
MODE LPT1: COM1:
```

The two lines shown above can be placed into the AUTOEXEC.BAT file by editing the file— Insert the two lines as shown, but use the correct values for your printer.

To activate the revised AUTOEXEC.BAT, type AUTOEXEC at the DOS prompt or re-boot the machine.

Printer Control Codes and Escape Sequences

Instructions to the printer concerning how to print a document, as opposed to the text to be printed, are called "control codes." Control codes are letters and numbers just like any other text, with one special exception. In order for the printer to tell the difference between a string of characters that represent a printing instruction and another string of characters that is merely to be printed, a special "alert" symbol must precede the control code characters. Herein lies one of the most confusing ideas ever to confront the reader of a printer manual. In reality, the notion is not difficult, but the terminology surrounding it can be perplexing.

We have no difficulty understanding that a string or series of characters to be printed could as well be called a "sequence" of characters. Nor is it difficult to comprehend that a string of characters representing an instruction to the printer might be called a "control sequence." The control sequences sent to a printer are in fact called "escape sequences." On very rare occasions, the escape sequence is entered by first pressing the ESC key on the keyboard, but most often it is not.

An escape sequence is just a sequence of characters that begins with a special character named "escape." Each of the characters (including digits and symbols) used on a personal computer has been assigned a unique number. Fair enough, since numbers are all that a computer can understand anyway. For the characters we are accustomed to dealing with (letters A-Z and digits 0-9, and punctuation marks), the numbering begins with 32. A "space" is 32; an exclamation point (!) is 33; a quotation mark (") is 34, and so on. You can see how this works by turning on the NUM LOCK key on your keyboard. Then hold down the ALT key and enter 33 on the numeric keypad. The result should be an exclamation point. (The numbers across the top of the keyboard will not work for this.)

What about the numbers below 32? They are reserved for identifying control codes. For example, a "line feed" is 10, and a "form feed" (to eject a page of paper) is 12. And, 27 is the number assigned to the special "Escape" character used to indicate the beginning of a printer control sequence (escape sequence). Now all we have to do is to look up the number for the control codes we would like to send to the printer, put 27 in front of them and off we go. Not quite. How is the printer supposed to know that 27 is "escape" and not just a plain old 27 to be printed? We need something else to make our "alert" symbol unique.

Commonly, the backslash (\) is used in front of the 27, which makes the "complete" escape character \27, or in some cases, a leading zero is used, thus: \027. (Note that the use of a backslash in printer control codes has nothing whatever to do with the backslash symbol as used to specify the root directory in DOS.). In dBase, the escape character does not use a backslash. Instead, it is specified as "chr(27)." (The quotation marks are not to be included with the escape character.)

Although the escape character is always ASCII "27," the exact characters used for the escape character code (e.g., whether a leading zero is required) and the control codes that follow it may vary from one type of printer to another. Further, since printer control codes can be imbedded in documents, spreadsheets and database programs, the user should refer to the printer manual or the application users guide to determine the exact syntax required.

HP Laserjet II Printer Control Codes

The character strings used by HP laserjet printers can be perplexing. They are explained in the manual provided with the printer, but there are a few things to watch out for. One is the difference between a zero (0) and a capital "O." The other is the difference between a one (1) and a lower case letter "l." Look closely at the required codes and be sure not to confuse these characters.

Here is an example of what an HP Laserjet escape sequence would look like in Lotus 1-2-3. By dissecting even one example it is hoped that the reader will be encouraged to explore the subject further.

In Lotus 1-2-3, the sequence is called the "setup string." It is accessed by entering /PPOS (i.e, Command-Print-Printer-Options-Setup) when a worksheet is on the screen.

When the spreadsheet is printed, the setup string is sent to the printer ahead of the spreadsheet, and thus "sets up" the printer. The setup string for landscape orientation and compressed print (16 characters per inch) is:

\027E\027&l1O\027(s16.66H

(Note that a leading 0 is used with the 27; this varies from one application to another.)

This strange looking string of characters is actually composed of three subgroups:

\027E   \027&l1O   \027(s16.66H

Here is what each of the foregoing sequences does:

\027E – "Escape E" clears the printer of its existing control codes (perhaps left over from a previous print command)

\027&l1O – This is the escape code (\027), plus the code that sets the printer to "Landscape" orientation (&l1O). The character following the ampersand (&) is a lower case letter "L," (l). The next character is a "one," (1). The final character is the upper case letter, "O," not a zero.

\027(s16.66H – This is the escape code (\027), plus the code that sets the printer to compressed print, (s16.66H. Note that the latter code begins with an "open parentheses" character, but there is no "closing parentheses" character -- this is not a misprint. The "s" must be lower case; an upper case "S" will not work. The 16.66 sets the print size (pitch). The final character is an upper case "H". A lower case "h" will not work.

Writing a Printer Setup Program in dBase

Here is what the sequences in the preceding paragraph might look like in dBase. The quotation marks and the "+" sign are to be included in the command, as shown. The two question marks (??) send the command to the printer. The resulting printout should be in landscape orientation and compressed print.

?? chr(27) + "&l1O"+ chr(27) + "(s16.66H"

To test this command, do the following:

1. Open dBase
2. At the dot prompt, type the command: SET PRINT ON [enter]
3. Type: ?? chr(27) + "&l1O"+ chr(27) + "(s16.66H" [enter]
4. Open a small database (so we have something to print):
   Type: USE <database name> (You provide the name) [enter]
5. Enter the command: LIST STRUC TO PRINT [enter]
   (to print out the database structure)
6. Type the command: EJECT [enter]
   (If necessary to eject the page)

Figure 16-6 is an example of a dBase III program for setting up the HP Laserjet printer. It is not offered as an example of elegant programming, but it does work. It may be keyed into a word processor such as WordPerfect and saved as an ASCII file having the name extension ".prg, for example, "SETPRINT.PRG." Copy the ASCII file into your working dBase directory. To run the program, at the dBase dot prompt type: do SETPRINT [enter]. A menu will appear, allowing you to select the orientation, margin and pitch.

Fig. 16-6. Example of a dBase Printer Setup Program

```
            PRINTER SETUP PROGRAM:  "PRINTSET.PRG"
* PRINTER CODES:
   *escape  =  chr(27)
   *portrait  =  "&l0O"
   *landscape =  "&l1O"
   *margin  =  "&a#L"
   *pitch   =  "(s#H"
* Note:  the "#" is to be replaced with the desired margin or pitch
*****************************
* Begin Program
* set choice to 0
CHOICE=0
* clear the screen
CLEAR
* write print menu as text display
TEXT
             SELECT PRINTER OPTION
         ====================
         Choice   Orient   Margin   Pitch
           1       Port      6       10
           2       Port     10       10
         ---------------------------------
           3       Port     10       16
           4       Port     20       16
         ---------------------------------
           5       Land      6       10
           6       Land     10       10
         ---------------------------------
           7       Land      0       16
           8       Land     20       16
         ---------------------------------
           9  Cancel printer setup program
         ---------------------------------
ENDTEXT
```

- 3??

```
* add blank line
?
* prompt for menu selection
  WAIT  "               Enter selection: "  to choice

*set printer to ON
SET PRINT ON

* execute printer command selected (CHOICE = 1 to 9)

DO CASE
*      Orientation      Margin      Pitch

CASE CHOICE="1"

*       Portrait          6           10

?? chr(27)+ "&l0O" + chr(27) + "&a6L" + chr(27) + "(s10H"
*set printer to OFF
SET PRINT OFF

CASE CHOICE="2"
*       Portrait          10          10

?? chr(27)+ "&l0O" + chr(27) + "&a10L" + chr(27) + "(s10H"
SET PRINT OFF

CASE CHOICE="3"

*       Portrait          10          16

?? chr(27) + "&l0O" + chr(27) + "&a10L" + chr(27) + "(s16.6H"

*SET PRINT OFF

CASE CHOICE="4"
*       Portrait          20          16

?? chr(27) + "&l0O" + chr(27) + "&a20L" + chr(27) + "(s16.6H"

CASE CHOICE="5"
*      Landscape          6           10
?? chr(27) + "&l1O" + chr(27) + "&a6L" + chr(27) + "(s10H"
```

398

```
CASE CHOICE="6"
*       Landscape           10             10
?? chr(27) + "&l1O" + chr(27) + "&a10L" + chr(27) + "(s10H"

CASE CHOICE="7"
*       Landscape           0              16
?? chr(27) + "&l1O" + chr(27) + "&a0L" + chr(27) + "(s16.6H"

CASE CHOICE="8"
*       Landscape           20             16
?? chr(27) + "&l1O" + chr(27) + "&a20L" + chr(27) + "(s16.6H"

CASE CHOICE = "9"
CLEAR
?? CHR(7)
@10,15 say "Printer program canceled."
SET PRINT OFF
SET CONSOLE OFF
* use counter to keep message on screen for a few seconds
   Delay = 600
   Do while delay>0
      delay = delay-1
   Enddo
CANCEL
SET CONSOLE ON
CLEAR
ENDCASE
clear
?? CHR(7)
@10,15 say "Setup has been sent to the printer."
SET CONSOLE OFF
SET PRINT OFF
* use counter to keep message on screen for a few seconds
   Delay = 600
   Do while delay>0
      delay = delay-1
   Enddo
SET CONSOLE ON
SET PRINT OFF
EJECT
CLEAR
* end of program
```

NOTES

GLOSSARY

AC.
Alternating Current. Electrical current that alternates (cycles) in polarity at a specified rate. The frequency of alternation is measured in cycles per second (CPS) or Hertz (Hz). CPS and Hz are the same unit of measure.

Access time.
The amount of elapsed time between the point when information (data) is called for, until delivery of the information is completed.

Adapter.
Any printed-circuit board or controller that plugs into a slot on the system board.

Address.
The logical location of a piece of data, other information or a set of instructions.

Ampere.
One ampere (amp). The unit for measuring electrical current. Amperes = volts ÷ ohms.

ANSI.
American National Standards Institute.

APA.
All Points Addressable. A mode of computer/display operation where all points of a display image can be controlled.

Archive bit.
One of the 8 bits in a file's attribute byte. The archive bit indicates whether or not the file has been modified (e.g., edited) since the last time it was backed up.

ASCII, Extended.
See Extended Character Set.

ASCII character.
Any character of the ASCII character set.

ASCII.
American National Standard Code for Information Interchange.

Asynch.
Abbreviation for "asynchronous." Pertains to communications in which the sending and receiving devices do not have to be synchronized by a timing signal, because the data frames being sent contain bits that indicate the start and stop point of each string of data bits.

Attribute byte.
A byte that indicates the attributes of a file, such as its being a hidden file, system file, read-only file, etc. Attributes can be set by using the DOS.

ATTRIBute command (DOS).
A command (in DOS 5.0) that allows the user to set the attributes of a file (e.g., archive, hidden, system, read-only).

AUTOEXEC.BAT.
An optional file on the boot diskette or hard drive containing instructions that will be executed automatically when the system is powered on or cold-booted up.

Backup disk.
A disk that contains information copied from another disk. Used to make sure that original information is not destroyed or altered.

Backup.
The process of copying (or backing up) a file or set of files onto magnetic or other media, for safekeeping.

Bad sector.
A defective disk sector. A sector not reliable for storing data.

Bank.
A group of associated memory chips forming a block of memory, readable by the processor (CPU) in a single cycle of the system bus.

BASIC.
A computer programming language: Beginner's All-Purpose Symbolic Instruction Code.

Batch file.
Any file with a file name extension of .BAT, often used to store and execute DOS commands. The file is run by typing its name at the DOS prompt and pressing the [ENTER] key.

Baud.
The number of times a data signal changes state each second. It is not synonymous with bits-per-second, but the two are roughly the same in PC communications applications.

Beep Code.
A series of audible beeps used to indicate error conditions, usually during the Power-On Self-Test (POST).

Bezel.
A cosmetic panel that covers or borders the face of a device such as a disk drive.

Binary.
Refers to the base-2 number system. Only the digits 0 and 1 are used in the binary system.

BIOS.
Basic Input/Output System. A set of programs to perform rudimentary (basic input/output) functions and handle communications between the computer and its peripherals. IBMBIO.COM contains additions and changes ("updates") to the BIOS in ROM on the system board.

BIT.
In binary notation, the smallest increment of data, represented by a 0 or 1. BIT is a contraction of the words, "Binary Digit."

Block.
A string of records or words, or a character string, to be treated as an entity.

Boot.
See Bootstrap.

Boot record.
A record on the boot disk that instructs the computer BIOS how to load the operating system files (e.g., DOS) into memory.

Bootstrap.
A procedure or method in which the execution of one step makes it possible to perform subsequent steps. For example, step 1 loads program X; then program X makes it possible to perform step 2, etc. From the vernacular, "to hoist one's self by one's own bootstraps." During the boot process, IBMBIO.COM is loaded first and, in turn, it loads IBMDOS.COM, which then loads COMMAND.COM. Thus, the entire process is referred to as "bootstrapping"— one file is loaded, which loads the next, which loads the next.

Boot diskette.
A diskette containing the files necessary to boot up the system.

Buffer.
A portion of RAM (user memory) used to store data while the data is being transferred from one device to another, i.e., a temporary storage area in memory.

Bug.
An error or defect in a computer program.

Bus.
An electrical pathway over which power, data and control signals travel.

Byte.
An associated set of 8 bits (usually to represent a character or digit).

Cache memory.
A reserved section of RAM in which data from frequently accessed disk/diskette sectors are stored. By thus storing the frequently used information, the number of disk accesses is reduced and operations are speeded up accordingly.

Capacitor.
A device formed of two plates separated by insulating material (which could be air), designed to store an electrical charge. The electrical charges accumulate on the surface of the plates, and are of opposite polarity across the plates.

Cathode Ray Tube.
An evacuated glass tube, one surface of which is coated with phosphor. Electron guns inside the tube shoot electrons against the phosphor, causing the phosphor to glow, and thus to create a visual display (e.g., as with a PC video monitor); a CRT.

Channel.
A path for transmitting signals.

Chip.
An integrated circuit (IC), usually within a small, rectangular case and mounted to a board by a socket or by soldering.

Circuit board.
A group of electrical circuits printed (etched) into the thin metal coating on a plastic sheet (board). Also called "Printed Circuit Board" and "Printed Circuit Assembly (PCA)."

Circuit.
A complete electronic or electrical path.

Cluster.
A group of one or more sectors that define an area on a disk where a piece of data is to be stored. Files are broken up and stored in one or more clusters; clusters are also called "File Allocation Units."

CMOS.
Complementary Metal Oxide Semiconductor. CMOS chips are used to store system configuration data, among other things.

CMOS RAM.
RAM used for storing configuration data and supported by a small battery so that the information is retained when the PC is powered off.

COMMAND.COM.
The command interpreter or user interface and program loader portion of DOS — the last file loaded during the boot process.

CONFIG.SYS file.
An optional (DOS) file on the boot disk; stores and loads device drivers, sets system buffers, and performs other rudimentary startup tasks. It thus "configures" the system.

Configuration file.
Any file maintained to store and execute the several conditions and parameters needed to run a particular software application or peripheral device.

Configuration.
The specific set of options (memory, adapters, displays, printers) that are installed on a PC.

Configure.
To set various hardware or software switches to indicate the various options and adapters that have been installed in a PC. Hardware switches are physical switches, found on the system board and on adapter boards. Software switches are not physical switches. Rather, they are logical conditions set by using a software utility program, usually by selecting an option from a program menu.

Console.
The keyboard.

Controller Card.
Any printed circuit board or card providing circuitry and components for controlling a device such as a disk drive. Also called an "interface" or "port." These cards are normally installed into one of the slots on the system board.

Controller.
Any assemblage of electrical and electronic circuitry and components that provide an interface between the computer and a peripheral device, such as a keyboard or disk drive.

Conventional Memory.
The area on the system where Random-Access Memory (RAM) may be installed between absolute addresses 0 and 640Kb; also called "user memory" or "base memory."

Coprocessor.
An optional microprocessor chip designed for processing floating point arithmetic and doing high speed calculations involving scientific formulas. Also called "Numerical Processing Unit (NPU)" and "math co-processor."

CPS.
(1) Cycles Per Second. The number of times per second that something changes its state or condition. In modern usage, CPS has been replaced by "Herz (Hz)." Both CPS and Hz express cycles-per-second. (2) Characters per second; usually the unit of measure for describing the speed of a printer or a display monitor.

CPU.
Central Processing Unit. The computer's main microprocessor chip. Also called a "microprocessor."

Crash.
A serious malfunction of the system wherein everything come to an abrupt halt, usually without warning; often applied to the failure of a disk drive due to mechanical problems (as opposed to recoverable software problems).

CRT.
See Cathode Ray Tube.

Cylinder.
A cylinder is a group of two or more tracks on a disk or set of hard drive platters. On a floppy disk that uses both its surfaces for data storage, each surface has its tracks numbered 0,1,2,3, etc. Thus, track 0 on the upper surface and track 0 on the lower surface together constitute cylinder 0. Track 1 on the upper surface and track 1 on the lower surface together constitute cylinder 1, and so on for the remainder of the pairs of tracks. On a hard drive having, for example, four platters ("disks"), with each platter having two sides, there would be a total of 8 tracks numbered 0, 8 tracks numbered 1, and so on. All of the tracks of the same number constitute the cylinder of the same number. For example, all of the track 0s together constitute cylinder 0.

Daisy chain.
Any arrangement of components in serial order, where signals flow from one component to the next (but not necessarily through the successive components). From the vernacular, suggesting that the components are connected in a fashion

similar to hooking daisies together to form a strand or chain. The cable for floppy drives is said to be a daisy chain because the cable runs from the adapter to the first drive and then on to the second drive. A Small Computer Systems Interface (SCSI) is configured as a daisy chain.

Data transfer rate.
The highest speed at which data can be transferred from one device to another.

Data.
Any group of facts, concepts, numbers, letters, or symbols, used as "information."

DC.
Direct current. Electrical current that flows at a constant polarity, in contrast to AC (alternating current), which changes its polarity 60 times per second (CPS, Hz) (or, 50 times per second, depending on the type of power provided in a given locale).

Default.
Any action, value or condition that the computer assumes to be applicable when no other instruction or command is provided to alter that assumption. For example, when the command "ERASE File_X" is entered at the DOS prompt and the directory wherein the file resides is not specified, the computer assumes that directory to be the one in which the user is currently working, i.e., the "default" directory.

Density.
The maximum amount of data that can be stored per unit area of a disk's surface. Disk density (capacity) is normally expressed as the number of Kbytes a disk will hold. Obviously, for a given amount of surface area, the more data that can be stored, the higher is the density of the data.

Device driver.
A software program that allows the computer to interact with a device, such as a printer or video display. Device drivers are usually loaded into memory via the CONFIG.SYS file. Once a device driver is loaded, it remains in memory, so that it can be called upon as needed. Drivers are terminate-and-stay-resident (TSR) programs.

Diagnostics Diskette.
Any diskette containing one or more programs designed to evaluate the status and performance of a computer, and/or to diagnose problems on the computer.

Diagnostics.
Any set of programs designed to test the various system components and aspects of performance, and to identify the source(s) of problems and malfunctions.

407

DIP (Dual In-line Pin).
Any chip mounted by means of two parallel rows of pins.

DIP (Dual In-line Pin) switch.
A switch mounted by pins arranged in two parallel rows. Usually, a DIP switch itself contains eight miniature switches. DIP switches are found on the system boards of early PCs and on adapter boards. DIP switches are used to configure the system and/or adapter board (i.e., to inform the system about the options installed).

Direct memory access.
The means by which data is written directly into system memory without passing through the CPU. This function is handled by a specialized processor chip.

Direct Memory Access (DMA) channel.
The pathway or channel assigned to a device for purposes of writing data directly from the device into system memory without the data passing through the CPU.

Directory.
A special file that holds the names of "working" files (programs and data – documents, spreadsheets, databases, etc.) along with the attributes (characteristics) of those files (such as "read only," "hidden," and the size of the file and the time and date the file was created). A directory also contains the names of any subdirectories beneath it, just as a file folder may contain another folder which in turn contains documents.

Disk Operating System (DOS).
A "master program" comprising several smaller programs which collectively allow the computer to run applications (e.g., WordPerfect; Lotus 1-2-3) and to communicate with peripheral devices such as disk drives, displays and printers. The term "disk" operating system (DOS) is misleading, in that DOS does far more than operate just the floppy drives and hard drives.

Disk.
Any disk-shaped medium, the surface(s) of which are used for storage and retrieval of information (e.g., computer programs or data).

Disk Drive.
The apparatus that rotates a magnetic disk and operates the read-write heads, such that data can be stored and retrieved on the disk's magnetic surface; also, the entire device, including the magnetic surfaces.

Diskette.
A small, magnetic disk (usually 5 1/4 or 3 1/2 inches in diameter). Because diskettes are thin and flexible, they have come to be called "floppy." A floppy disk is a diskette. This is in contrast to a rigid disk inside a hard drive, which is called a "platter." The smaller (3-1/2 inch) diskettes are housed in a rigid plastic jacket, but they are nonetheless "diskettes," or "floppies."

Display.
Typically, a video or cathode-ray tube device used to provide a luminous representation of information; a monitor. Other types of displays are formed of light emitting diodes (LED's) or liquid crystal, both of which form patterns by altering the luminous or reflective properties of the display's surface.

Distribution Diskette.
Any diskette, the purpose of which is to convey (transport) a program (e.g., an application) from the author or manufacturer to the user.

DMA.
See Direct Memory Access.

DOS diskette.
A diskette containing the following files: DOS COMMAND.COM; IBMBIO.COM; and IBMDOS.COM. A PC can be booted up by a DOS diskette in the machine's (A:) floppy drive.

Edge connector.
A series of flat, electrical contacts that form an integral part of the edge of a circuit board. The edge protrudes such that the board can be plugged into a slot.

EISA.
Extended Industry Standard Architecture. An improved and extended/expanded version of the Industry Standard Architecture (ISA) upon which the early PCs (IBM PC, XT, AT, and clones) were based. EISA is the non-IBM industry's proposed competitive answer to IBM's MicroChannel Architecture (MCA).

Electronic disk.
Not a disk at all; rather, it is an area of memory that has been set up to act like a disk. It has the advantage of speed, because memory is much faster than a mechanical disk drive. It has the disadvantage of loosing all of its data when the power to the PC is turned off.

Error code. A number that represents an error or malfunction in the computer, usually displayed on the screen (if the screen itself is not the problem).

Error signal.
An audible sequence of beeps (tones) produced by the speaker in the PC, to indicate "O.K." status, or errors in the system. Different combinations of long and short beeps represent different error conditions, in a fashion of coding similar to Morse Code.

Error message.
A string of words on the display that supposedly tells you what is wrong, but which are generally too vague to be of immediate help.

Escape code.
A string of characters preceded by the ASCII "ESCAPE" character (ASCII 27); also called an "escape sequence." These sequences are used to distinguish between ordinary text and instructions to be followed by a device, such as a printer. When dealing with printers, an escape sequence is also called a "setup string," (e.g., as in the Lotus 1-2-3 printer-setup menu).

ESDI (Enhanced Small Device Interface).
An improved disk interface system, faster than the "standard" ST-506 system.

Expanded memory.
Random-access memory installed on a device that complies with the Lotus/Intel/Microsoft (LIM) Expanded Memory Standard.

Expansion memory.
Any random-access memory (RAM) installed to expand the conventional (user) memory capacity of a PC. Not to be confused with "Expanded Memory."

Extended memory.
The contiguous block of random-access memory starting immediately above the first 1Mb (1024Kb) of RAM. Contiguous means that the addresses of the installed units of memory are in sequence, with no gaps between them.

File Allocation Table (FAT).
A table stored near the outer edge of a disk that identifies the sectors to which the files are allocated (i.e., the sectors in which files are stored). The root directory stores the identification number of the first sector for each file; the FAT table then stores the identification of the remaining sectors, with each entry in the table pointing to the next location of the next sector for that file. The FAT thus represents a chain of sectors that stores the entire file. The FAT also identifies sectors that are available for use. DOS maintains two copies of the FAT on the hard disk. The second FAT is not immediately available to the user, but it can be accessed to restore a damaged FAT by using the Norton Utilities.

File. Numerical data, text, graphics or computer instructions forming an entity and having a name by which it may be stored and retrieved.

Fixed disk.
Data storage using magnetic, optical or other medium disks, where the disks are sealed inside the drive mechanism and cannot be removed without disassembling the device; also called a "hard disk," in contrast to floppy disks (flexible diskettes), which are removable from their drive.

Floppy disk.
A flexible, removable disk. *See* Diskette.

Format.
To prepare a disk surface such that data can be written to it. *See* format, low-level; and format, high-level.

Format, low-level.
The process and end result of magnetically writing tracks and sectors onto a hard disk, setting the interleave factor, and writing the headers (sector descriptors) into each sector. On a previously unformatted hard disk, low level formatting must be done before high level formatting is performed by the operating system. On a floppy disk, DOS performs both low-level and high-level formatting as a single process.

Format, high-level.
The process and end result of mapping out the sectors to be used for the boot record, file allocation tables and root directory, and for writing the COMMAND.COM and hidden files (IBMSYS.COM; IBMBIO.COM) to the diskette or hard disk.

FORMAT.COM.
The DOS program used to high-level format a hard disk, and to perform both low- and high-level formatting of a floppy disk.

Formatted capacity.
An unformatted disk theoretically can hold a certain amount of information (data). But, the processes of low-level formatting and high-level formatting use up some of that capacity, because the format is written to the disk much as data is written to the disk later on. Therefore, the "unformatted capacity" is reduced by the amount of space taken up by the format. The space that is left is the "formatted capacity," which expresses the number of Kbytes or Mbytes of "user" data the formatted disk can hold.

Formatting.
The process and end result of preparing a disk to receive data; also referred to as "initializing." *See* low-level formatting; and high-level formatting.

Hard Disk Drive.
See Fixed Disk.

Head parking.
When a disk drive is powered off, its read/write heads come to rest on the surface of the disk. This is not good for the heads or for the disk. Mainly, the problem is that it risks destroying data if the heads land on a track that contains data. So, a procedure (a program) called "Park Heads," or "Prepare [machine] for Moving," is available that moves the heads onto an unused track (a track outside of those used to store data, boot the system and format information).

Head crash.
A mechanical malfunction or failure in which the heads of a disk drive physically impact the magnetic data storage surface of the disk, thus destroying the data stored there. (*See* crash.)

Head.
An electromagnetic component inside a drive that imparts magnetic charges to the disk surface (i.e., "writes" to the disk); senses magnetic charges on the surface (i.e., "reads" from the disk); and "erases" data by writing illegible magnetic charges into the areas specified for erasure (or otherwise "deletes" files by altering the first character of the file name, thus making the space occupied by the deleted files available for occupancy by another file).

Hex.
Hexadecimal.

Hexadecimal number (Hex number).
A number in the base-16 system of counting. Hexadecimal includes the characters 0-9 and A-F, for a total of 16 characters. Every hexadecimal number can be expressed in terms of its decimal equivalent. For example, hexadecimal "A" is equivalent to decimal 10; hexadecimal F is equivalent to decimal 15.

Hidden file.
A file is hidden in the sense that, even though the file is present on the disk, its name is not displayed in any directory listing in response to the DOS DIR (directory) command. Files may be hidden or "un-hidden" (revealed) using the DOS ATTRIBute command.

High Memory Area.
The first 64Kb of RAM the extended memory area.

High-level formatting.
See Formatting, high level.

IBMBIO.COM.
This is one of the DOS system files needed in order for the system to boot up. It contains extensions to the BIOS stored in ROM on the system board. *See* Bootstrap.

IBMDOS.COM.
This is one of the DOS system files needed in order for the system to boot up. It is loaded by IBMBIO.COM. When IBMDOS.COM is loaded, it in turn loads COMMAND.COM.

Initialize.
See Format.

Interface.
A device, procedure, software program or other means whereby two unlike devices are able to communicate with each other. By analogy, if person A speaks only French, and person B speaks only Spanish, they could communicate through an interpreter (an interface), provided that the interpreter could speak both languages, and thus translate French into Spanish, and vice versa. Certain interfaces, however, need work in only one direction, such that the output of the first device is made understandable as input to the second device.

Interleave factor.
The sectors on a disk are read in numerical order. Since disks often spin faster than they can be read, the sectors are numbered out of sequence as a form of compensation. The intervening, out-of-sequence sectors pass by, thus giving the read head time to recover before the next sector arrives to be read. The number of intervening sectors plus 1 is the "interleave factor."

Interrupt.
The temporary cessation of a process, caused by an event external to that process. It is in fact an "interruption." But in computer parlance, an "interrupt" is a formal process whereby the process that gets interrupted can remember where it was when it was interrupted, and thus can later resume its activity from that point, if necessary to do so.

ISA (Industry Standard Architecture).
The system architecture introduced in the first IBM personal computer (PC) and subsequently used in the XT and AT.

Jumper.
A tiny connector that slips over two pins that protrude from a circuit board. When in place, the jumper connects the pins electrically. By doing so, it connects the two terminals of a switch, turning it "on."

Kbytes and kilo.
The uppercase "K" is understood to represent 1024. This should not be confused with lowercase "k," which is the abbreviation for kilo or thousand, exactly. Thus, 64Kb is actually 64 x 1,024, or 65,536 bytes, not 64,000 bytes.

Kilobyte.
One kilobyte equals 1,024 bytes.

Landing zone. A track (not used for data) on a disk surface where the read/ write heads can land on when power is shut off. This is the place where a parking program or a drive with an autopark mechanism will park the heads.

LED (Light-Emitting Diode). A special-purpose semiconductor device used as an indicator (power on, device-in-use, etc.) lamp.

LIM (Lotus/Intel/Microsoft) Expanded Memory Standard (version 3.20, September 1985).
A specification for expanding the memory-addressing capability of Intel microprocessors in IBM personal computers and their clones.

Low-level formatting.
Formatting that divides tracks into sectors on the disk or platter surfaces.

Mbytes and mega.
One Megabyte (1Mb) is equal to 1024 x 1024 bytes, or 1,048,576 bytes. In contrast, the term "mega," (when abbreviated using a lower case letter "m") represents one million (1,000,000).

MCA (Micro Channel Architecture).
The bus architecture system introduced by IBM in its PS/2 system (models 50-80).

MCGA (Multi-Color Graphics Array).
The video system built into the IBM PS/2 models 25 and 30. It supports the old IBM CGA (but not EGA) modes, plus VGA modes 11 and 13.

Memory.
Electronic "workspace" (for temporary storage of computer instructions and data) provided mainly by semiconductor chips, either RAM or ROM chips. RAM chips lose any information they may be holding when the power to the PC is turned off. ROM chips retain their information in the absence of PC power, but the PC user cannot write information into these chips. ROM chips have their information inserted by a special process not available to the average PC user.

MEM.
In DOS 5.0, MEM is the command that will display the amount of memory (RAM) installed on the system.

Memory-resident program.
A program that stays in memory (RAM) once it has been loaded. Once the program is loaded, it may be activated at any time, even during another program. *See* Terminate-and-Stay-Resident (TSR).

MHz.
One MHz (MegaHertz) equals one million cycles per second.

Microsecond.
One millionth of a second.

Millisecond.
One thousandth of a second. Abbreviated as "ms."

Modem.
A contraction of the words MOdulator/DEMmodulator. A modem converts (modulates) digital electrical signals from a computer into analog signals that can be transmitted over telephone lines to another computer. The conversion from analog back to digital form, so that the receiving computer can "understand" the transmission, is called "demodulation."

Motherboard.
The main circuit board of a personal computer; also called "system board."

Multitasking.
The process of running several programs at the same time.

Overlay.
Part of a computer program that is loaded into memory only when it is needed.

Overwrite.
To write data into a location where data already resides, thus eradicating the data previously written to that location and replacing it with the newly written data.

Parallel transmission.
A process whereby data and/or control signals are sent (transmitted) "side-by-side," rather than one after the other, as they are in serial transmission.

Peripheral.
Any item of equipment attached to a computer that is not part of the main computer system, e.g., a printer, disk drive, modem or display.

Pitch.
The number of characters per inch printed on a horizontal line. For example, "12 pitch" type indicates 12 characters per inch.

Platter.
A disk inside a hard-disk drive.

Point.
A point is 1/72 of an inch.

POST (Power-On Self Test).
The series of tests that a personal computer performs on itself whenever the machine is powered on.

Power supply.
An electrical and/or electronic device that provides the needed operating voltage and current to the computer. A PC power supply normally converts high voltage AC power into low voltage DC power.

Processor speed.
The clock rate at which a CPU processes data. Depending on the make and model, personal computer clock rates range from 4.77 MHz to 50 MHz at the time of this writing.

Prompt.
A symbol presented on the display screen indicating that the system is ready to receive a command from the keyboard. In DOS, the prompt is the character ">," accompanied by a blinking cursor.

416

RAM disk.
A disk "simulated" by the use of random access memory (RAM). The RAM acts just like a physical disk, except that the RAM disk contents will be lost when the power to the PC is turned off, unless those contents have been copied to a real (physical) disk. A RAM disk is also called a "virtual disk," or an "electronic disk."

RAM (Random-Access Memory).
Memory into which the user may write instructions and data.

Read-only file.
Any file which can be read but not written to or erased. In DOS, this condition is controlled by the file's attribute byte which, in turn, is set by using the DOS attribute (ATTRIB) command.

Read-Only Memory (ROM).
Memory whose contents can be read but which cannot be altered by ordinary PC commands. ROM is used mainly to store information and instructions needed by the machine when the machine is powered up.

Reserved memory.
The area of memory area between 640Kb and 1024Kb.

ROM BIOS.
The Basic Input-Output System of the computer, which is stored in a read-only memory (ROM) chip on the system board.

ROM (Read-Only Memory).
Memory whose contents may be read but not altered.

Root directory.
The main directory of a disk. The root directory is created during the DOS FORMAT process. All other directories created under DOS are, by definition, subdirectories of the root directory.

Scratch disk.
Any floppy diskette that contains no information of value. The disk thus can be used safely for testing disk drives. Note that certain tests may require a specially formatted "scratch disk."

SCSI (Small Computer System Interface).
An interface system between the personal computer and add-on devices such as hard disk drives, CD-ROM drives and printers. The SCSI uses a "daisy chain" arrangement for connecting the devices, in contrast to individual adapter slots for the devices.

Sector.
A section of a single track of a disk, comparable in function to a data cell on a spreadsheet. Each sector on the disk is numbered; a sector typically holds 512 bytes of data.

Serial Transmission.
The process of transmitting or transferring data one bit at a time. Contrast to "parallel," which sends 8 data bits concurrently.

Shadow RAM.
When information stored in ROM is written to RAM and used from its RAM location (address), the RAM is said to "shadow" the ROM, in that it functions the same as the ROM. Shadow RAM is used because it is faster than working directly with ROM-stored information. When shadow RAM is created and used, the corresponding ROM is meanwhile disabled.

Shell.
A "shell" is any software by which the user interfaces with the system. For DOS, the shell is the program "COMMAND.COM." COMMAND.COM is the program by which user commands entered at the keyboard are "translated" for use by the system.

SIMM (Single Inline Memory Module).
A plug-in, circuit board on which RAM chips are mounted. The board plugs into a socket on the system board (motherboard) by means of an edge connector.

Source Disk.
A disk containing information to be copied to another disk. The disk receiving the information is the "target" disk.

Spindle.
The post to which the platters of a hard disk drive are affixed (mounted).

ST-506/412.
The standard interface for hard disk drives in IBM personal computers.

Standby power supply.
A backup power supply that switches into operation when the main (line) power fails. Because it must switch into operation, a standby power supply is not an "uninterruptable power supply," regardless of how fast the switching occurs.

String.
A sequence of characters.

Subdirectory.
A directory listed within another directory. A subdirectory in DOS is essentially a special file that holds the names and characteristics of other files such as applications program files and user-created files (documents, data).

Surge protector.
A device that prevents sudden increases in voltage (spikes) in the line power (e.g., from the wall outlet) from reaching the computer. In function, a surge protector is similar to a fuse.

Synchronous.
A method of data transmission in which the sending system and the receiving system are coordinated (synchronized) by a timing signal.

System Files.
The files that must be present on a system diskette: IBMDOS.COM, IBMBIO.COM and COMMAND.COM. On non-IBM systems, these files will have different names, but their functions will be the same.

System Diskette.
A diskette containing a copy of an operating system. In DOS, a system diskette must contain IBMDOS.COM, IBMBIO.COM and COMMAND.COM. Another name for system diskette is "boot disk."

System Board.
The main printed-circuit board within the computer. It contains the system's base memory, central processing unit and numeric processing unit (NPU), if an NPU is present. The system board also contains slots for plugging in adapters to support the various "add-on" devices such as disk drives, displays and printers; also called "motherboard."

Track.
Any of the concentric circles on a storage medium (such as a magnetic disk) onto which data may be written. Each track is divided into sectors to form specific areas for data storage. Typically, each such area (sector) holds 512 bytes of data.

TSR.
Terminate-and-Stay-Resident. A TSR program is any program that remains in RAM after it is loaded, regardless of whether the program is used or not; also called a "memory-resident" program.

Unformatted capacity.
The total amount of data (number of bytes) that a disk could hold, if all of the space on the disk were available for data storage. But because part of the space

419

is used in formatting the disk, the total capacity for data storage is reduced. The space remaining for data storage after the disk is formatted is the "formatted capacity" of the disk.

Uninterruptable power supply (UPS).
A device that provides operating power from batteries; the batteries are kept charged by the line (i.e., wall outlet) power. When the line power (main power) fails, the batteries continue to provide power, without interruption, until they become discharged. Since no switching is needed when the main power fails, the supply is in fact uninterruptable. Contrast to "Standby (switched) power supply."

VGA (Video Graphics Array).
The video subsystem of the IBM PS/2 family of personal computers. VGA emulates all previous video modes and introduces several higher resolution modes.

Video Memory.
The 128-Kbyte area immediately above conventional (user) memory; also called "graphics memory."

Virtual memory.
A technique whereby a disk drive is used to serve in the same role as memory. Programs and data (or portions thereof) are kept on the disk and swapped back and forth into system memory as they are needed.

Virtual Disk.
See RAM Disk.

Wrap-plug.
A device that plugs into a communications port for purposes of testing the port. The wrap-plug receives output from the port and echoes it back into the input side of the port. The functioning of the port is evaluated by comparing the output to the input. They should be the same.

Write-precompensation.
The method of adjusting the timing and current of the disk write-head, to compensate for the closer spacing of data written to the inner tracks (cylinders) of a disk. Compensation is necessary because of the magnetic influence between adjacent bits on the magnetic surface, which causes the bits to shift (move or drift).

Write.
To record data to a disk or to random-access memory (RAM).

REFERENCES

[1] Minasi, Mark. *The Complete PC Upgrade and Maintenance Guide.* Alameda: SYBEX, Inc., 1991.

[2] Mueller, Scott. *Upgrading and Repairing PCs.* Carmel: Que Corporation, 1988.

[3] Woram, John. *PC Configuration Handbook.* New York: Bantam Books, 1990.