Introduction

The primary purpose of this book is to prepare you to troubleshoot and repair microcomputer systems and their peripherals.

This goal is achieved through a three-part effort:

- Solid theory presentation
- Hands-on operation and exploration in lab experiments
- Troubleshooting applications in the lab procedures

Initially, solid theory thoroughly describes typical microcomputer architectures and operations. In each case, underlying topics are presented as concepts, advanced through actual applications, and presented as they are actually implemented in a typical PC.

Secondly, the lab procedures give you a practical, hands-on example of these theoretical concepts through actual experimentation with a live system. Most of these labs are on the CD-ROM included with the book.

Finally, the book develops the most advanced learning levels by causing you to analyze the system when faulty components are installed. In these situations, you must understand how the system is supposed to operate as well as how various malfunctions could alter its operation.

Note

There are 37 lab procedures included with this book on the CD-ROM. They offer a variety of presentation styles. A number of software-oriented labs are included. These labs include introductions to DOS, Windows, Windows 95, and QBASIC. Advanced software exploration labs in Windows, Windows 95, and QBASIC are also included. Different software diagnostic tools are covered. These include Microsoft's MSD, PC-Check, and CheckIt diagnostic software packages.

The CompTIA organization has established objectives (refer to Appendix C, "A+ Objective Map,") for the core portion of the A+ Certification examination.

How This Book Is Organized

In general, it is not necessary to move through this text in the same order that it is presented. Also, it is not necessary to teach any specific portion of the material to its extreme.

Chapter 1—"Microcomputer Fundamentals," introduces basic microcomputer architecture by presenting organizational and operational issues associated with a typical microcomputer system. The chapter includes information about the operation of a simple mythical microprocessor to illustrate microprocessor fundamentals.

Chapter 2—"PC Hardware," builds on the fundamental material from Chapter 1 to show how those basic microcomputer structures come together to form an IBM PC-compatible personal computer system. The chapter charts the evolution of the PC from the days when small keyboard units were connected to a television set up to the basis of the most powerful PCs available today.

The software side of the microcomputer system is covered in Chapters 3 and 4.

Chapter 3—"Operating Systems," provides a detailed examination of basic operating systems. In particular, it investigates the role of the system's BIOS and the disk operating system (DOS) in the operation of the system. The first half of the chapter covers topics associated with the ROM BIOS. This includes system bootup information, CMOS Setup routines, and Power On Self-Test information.

The second half of the chapter is dedicated to DOS. Topics covered in this section include installing and starting DOS systems, using different types of DOS commands, and using DOS utility programs.

Chapter 4—"Microsoft Windows," is an in-depth study of the Windows 3.x operating environment and the Windows 95 operating system. This chapter looks at installation, startup, structure, and operation of Windows 3.x and Windows 95 system software.

Chapter 5—"Troubleshooting the System," addresses the fundamentals of troubleshooting microprocessor-based equipment. The chapter covers the use of diagnostic software to isolate system problems. It also describes the field replaceable unit (FRU) method of hardware troubleshooting required for most field and bench repair work. An extensive section is included for troubleshooting DOS, Windows 3.x, and Windows 95 problems.

Chapter 6—"System Boards," deals with the system boards that make up the heart of every microcomputer system. Microprocessors, microprocessor support systems, and expansion buses are covered in this chapter. These support systems include timing, DMA, interrupt, common memory structures, and different I/O bus schemes used to connect optional I/O devices to the system.

In addition, the chapter covers microprocessors from the 8088 to the Pentium, Pentium MMX, Pentium Pro, and the Pentium II. The operating characteristics of all these Intel microprocessors are presented in this chapter. Procedures for troubleshooting system board-related problems and system board upgrading are also presented here.

Chapter 7—"Input/Output," begins to examine basic input/output systems. In this chapter, the computer's fundamental I/O devices are covered. The most common ports found in the

PC-compatible world are the parallel and serial I/O ports. However, a host of lesser known I/O systems are used in PCs. The various I/O systems are examined here. The chapter also investigates the operation of various input devices. In particular, keyboards, mice, trackballs, joysticks, light pens, touch-sensitive screens, and scanners are covered. Troubleshooting procedures are provided for the various devices.

Chapter 8—"Magnetic Storage," presents mass storage systems commonly used in microcomputers. These include floppy drives, hard drives, RAID systems, and tape drives. Installation and troubleshooting procedures are described for the hard and floppy drive systems, as well as for tape drives.

Chapter 9—"Video Displays," investigates the video display area of I/O devices. Basic CRT construction, operation, and control are featured. The various video standards associated with microcomputers are addressed along with troubleshooting procedures for isolating video problems. The end of the chapter deals with troubleshooting the CRT monitor. Readers should be aware that this section contains information about a potentially dangerous portion of the system and should be practiced only with a trained professional.

Chapter 10—"Printers," covers, obviously, printers. From the workhorse dot-matrix printer to color ink-jet and high-speed laser printers, theory, operation, and maintenance information is presented for all three types of printers.

Chapter 11—"Data Communications," focuses on one of the hottest areas of microcomputer growth. This chapter covers both local and wide area networks along with the equipment and software required to operate them. Modems are covered here along with their application to the Internet. Procedures and precautions for troubleshooting networked systems is provided.

Chapter 12—"Multimedia," investigates the components that make up true multimedia systems. Another hot area of microcomputer development is occurring in the area of multimedia production and use. These items include CD-ROM drives, sound cards, video capture cards, MIDI sound equipment, and the software required to drive these items. Troubleshooting directions are included for most multimedia systems.

Chapter 13—"Preventive Maintenance and Safety," discusses important preventive-maintenance procedures and safety considerations. The first section of the chapter investigates topics including cleaning, electrostatic discharges, power-line problems, universal power supplies (UPS) are discussed, along with other power-line conditioning devices.

The midsection of the chapter features preventive-maintenance procedures for various system components. Important HDD support utilities such as backup, defragmentation, and antivirus protection are highlighted here. Suggested PM schedules are also presented.

The final section of the chapter involves safety issues concerning computer systems. Although not an intrinsically unsafe environment, some areas of a computer system can be harmful if approached unaware.

Appendix A completes `A+ Certification Training Guide` by examining issues of customer satisfaction. Traditionally, these issues have not been covered in technical books associated with computer maintenance. However, increased demand from computer service providers has finally brought the topic into the spotlight. Many service providers consider the ability to handle customers and their problems effectively as important a skill as having good technical abilities.

Appendix B is a thorough glossary that will help you understand key terms while you read this book.

Appendix C contains the A+ Objective Map.

Test Taking Tips

The A+ exam is an objective-based, timed test. It covers the objectives listed in Appendix C in a multiple-choice format. There are two general methods of preparing for the test. If you are an experienced technician, using this material to obtain certification, use the testing features at the end of each chapter and on the accompanying CD-ROM to test each area of knowledge. Track your weak areas and spend the most time concentrating on them.

If you are a newcomer to the subject of serious computer repair, plan a systematic study of the materials, reserving the testing functions until each chapter has been completed.

In either case, after completing the study materials, use the various testing functions available on the CD-ROM to practice, practice, practice, taking the test. Test yourself by topic, as a mixture of questions form all areas, as a flash card review, and so on until you are very comfortable that you are ready. The CD-ROM will enable you to immediately reference the area of the text that covers material you might miss.

Don’t forget the following:

- Answer all the questions you know first. You can always go back and work on harder questions.
- Don’t leave any questions unanswered. They will be counted as incorrect.
- There are no trick questions. The most correct answer is there somewhere.
- Be aware of A+ questions that have more than one correct answer. They are identified by a special formatting of the letters for the possible answers. They are enclosed in a square box. When you encounter these, make sure to mark every answer that applies.
- Get plenty of hands-on practice before the test. Practice against the time limit set for the test.

- Make certain to prepare for each test category listed in Appendix C. The key is not to memorize, but to understand the topics.
- Take your watch. The A+ exam is a timed test. You will need to keep an eye on the time to make sure that you are getting to the items that you are most sure of.
- Get plenty of rest before taking the test.

Acknowledgments

I would like to mention some of the people and groups who have been responsible for the success of this book. First I would like to thank Greg Michael, formerly of Howard W. Sams, for getting me involved in writing about microcomputer systems back in the early days of the IBM PC.

Several people have given me valuable input for the disk drive chapter. First there was David Allan, formerly of Tallgrass Technologies, makers of high-speed backup systems. Further information has been provided by Michael Arkin, Steve Montero, and Brett McClelland of Silicon Systems. They make ICs for disk and CD-ROM drive applications.

My staff here at Marcraft makes it easy to turn out a good product. Thanks to Paul Haven, Wanda Dawson, Renia Irvin, and Allen Hoy from the technical services area for trying things out for me. Also, Mike Hall, Whitney Freeman, Cathy Boulay, and Yu-Wen Ho from the Marcraft product development department for their excellent work in getting the text ready to go.

In addition, I would like to say thanks to Brian Alley and Melissa Ryan of Boston University for their excellent guidance in bringing this version of the book up to speed. I must also say that it has been a pleasure to work with Nancy Warner and Steve Weiss from New Riders Publishing.

As always, I want to thank my wife Robbie for all of her support and help with these projects.

Thanks to my son Robert also. He's given up a lot of "dad time" to let me write these books.

About Marcraft

Marcraft has been on the cutting edge in providing programs to prepare students for occupations in technical career fields since 1975. Marcraft was founded expressly for the purpose of designing, developing, and marketing electronic training systems for technology education.

We stay abreast of technological advances and continually improve our products to provide the best technology-based education products available. The company's products have been adopted in a wide range of training environments, including middle schools, junior-high schools, high schools, community colleges, universities, and private training groups.

Marcraft produced its first microcomputer architecture and repair course in 1988. In those days, computer repair was normally carried out by electronic technicians and engineers. That course was very hardware intensive, using chip-level isolation and repair techniques. Schematic diagrams and electronic test equipment was the order of the day.

The tasks performed by computer repair technicians has changed considerably since the original course was introduced. Computer hardware has become relatively inexpensive and software has become much more complex. Current computer technicians spend much more time dealing with software-related problems, configuration problems, and compatibility problems than with hardware problems. As a matter of fact, hardware problems in microcomputers are typically solved at the board level these days. The time and expense of an IC-level repair quickly goes beyond the value of the board, making it unprofitable to do these kinds of repairs.

Appropriately, Marcraft's `Microcomputer Systems - Theory & Service` course has changed several times to reflect the evolving nature of the technician's role. The courseware is in its seventh edition.

At the same time that Marcraft has evolved its training system to match the requirements placed on service personnel, the computer industry has established certification criteria for this same group. The organization is the `Computing Technology Industry Association` and the certification is called **A+**.

Marcraft has embraced the A+ certification because it sets a standard for excellence that we have been preparing students for from the beginning. Therefore we are offering this book to prepare students to successfully challenge the A+ examination. From its size, one should gather that this title is not just a cram course for the test. Instead, it is a complete training course designed not only to prepare for the exam, but also to provide the fundamental knowledge base required to establish a career in this rapidly changing industry.

Chapter 1

Microcomputer Fundamentals

Upon completion of this chapter and its related LabExercises, you should be able to:

Objectives

- Define the term "computer word" and describe the three types used in digital computers.
- State common word sizes used in digital computers, and relate the terms bit, byte, and nibble to word size.
- Describe any number using binary, hexadecimal, BCD (Binary-Coded Decimal), and octal notation.
- Convert any number from a given number system to any other number system.
- Given an ASCII code table, express any set of characters as 1s and 0s.
- Describe commonly used instruction word formats.
- Define the terms hardware, software, and firmware.
- Explain the functions associated with each of the four fundamental blocks that form a computer.
- Differentiate between common primary and secondary memory devices or systems and list examples of each.
- List various types of Integrated Circuit ROM devices and describe the differences between them.

continues

- Describe four common ROM applications.
- Differentiate between static and dynamic RAM memory devices, and state the conditions that dictate which of the two is used in a given application.
- List the three major buses in a computer system and describe the nature of the information each bus carries.
- Demonstrate the relationship between the size of the computer's address bus and the number of individual memory and I/O locations that it can access.
- Use "K" notation as it applies to addressing.
- List control bus signals that are found in most microcomputers.
- Explain the function of a microprocessor's instruction set.
- List the steps that occur during an Instruction Cycle of a typical computer.
- Define the functions of the computer's Input/Output Units.
- Contrast the four most common methods of initiating an I/O-Memory or Memory-I/O data transfer (Polling, Interrupts, DMA, and programmed I/O).
- Describe the events that occur when an interrupt signal is generated.
- Describe the events of a typical DMA operation and differentiate between the different modes of DMA transfers.
- Explain the three classes of software associated with computer systems.
- Explain the function of the system ROM BIOS.

- Describe the function and purpose of Disk Operating System.
- Describe the value of a Graphical User Interface.
- Describe popular software applications programs.

Introduction

This chapter examines the fundamental operation and organization of digital computer systems. It lays the foundation for the various in-depth topics covered in subsequent chapters.

The first section of the chapter describes what it is that digital computers actually understand—binary language and logic.

The focus shifts in the second section to cover the basic hardware building blocks that make up a digital computer system. These building blocks include microprocessors, microprocessor support systems, memory units, and buses.

This section is followed by a discussion of how the parts of the computer communicate with the outside world. The four common methods of conducting data transfers are defined, followed by typical input/output hardware conventions and systems.

The final section of the chapter introduces the topic of software. The first portion of the section deals with the system software that controls the operation of the hardware while the second portion introduces several common application software categories that enable users to create and manipulate different types of data.

Bits, Bytes, and Computer Words

Objectives

The basic structure of most computers in existence today is composed of digital devices. Digital devices, and therefore digital computers, process electrical signals that can assume only two possible states. These states correspond to a prescribed `high level` of voltage (such as +5.0V dc) and another `low level` of voltage (such as +0.5V dc). In digital electronics, these voltage levels are referred to as `high` and `low` `logic levels`, or as "1" and "0" `logic states`. Because the computer is an electronic device, these voltage levels are all that it recognizes. However, a number of conventions have been developed to enable humans to more easily relate to the logic understood by digital computers.

The logic values used in the digital computer correspond directly to the digits of the binary, or Base-2, number system. In this system, each digit can assume only two possible values: 0 and l. This basic unit of information is referred to as a bit (a contraction of binary and digit). Very little useful information can be conveyed by a single bit; virtually any number, letter, or symbol, however, can be described by organizing a group of related bits into coded bit patterns called words.

Within the computer, three types of computer words exist:

- Pure binary data words
- Coded data words
- Instruction words

These classifications describe the type of information the word conveys. These classifications are discussed in the sections that follow, but another aspect of computer words must be covered first—their length.

Word Length

Objectives

Word length is so important to a computer that it is often used to describe the computer, because all its internal hardware devices are constructed to accommodate a given word length. In general, computers that use larger word sizes are more powerful than those with smaller word sizes. This is because more information can be conferred at one time by a larger word size. Indeed, the increased number of possible bit patterns provided by larger words allows both a larger range of numeric values, and a greater number of characters and symbols that can be specified.

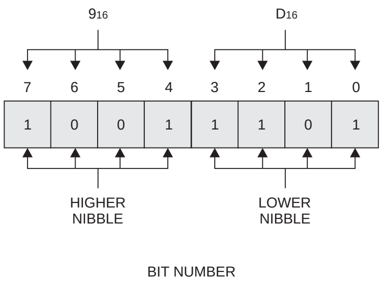
A number of different word sizes are commonly used with digital computers, but the most common is the 8-bit word, or byte. The term byte is so universally accepted in the computer world that even computers with other word sizes, such as 16- or 32-bit word sizes, are described in terms of their number of bytes. A computer using 16-bit words is said to have a two-byte word size. The word is divided into a higher- and a lower-order byte. The lower-order byte consists of the 8 least significant bits (LSBs, bits 0–7) and the higher-order byte is made up of the 8 most significant bits (MSBs, bits 8–15). Notice that the least significant bit is always assigned the bit number 0.

It is also common to divide bytes into two 4-bit sub-units called nibbles. Therefore, the byte consists of both a higher- and a lower-order nibble. This relationship was contrived largely due to the link between the binary (base-2) and hexadecimal (base-16) numbering systems. A 4-bit binary pattern (nibble) can be easily converted into a single hexadecimal digit, which is much easier to manipulate and communicate than a string of 1-s and 0-s. In this manner, a 2-byte computer word can be expressed more simply as a 4-digit hexadecimal word. Figure 1.1 depicts the relationship that exists between binary and hexadecimal numbers.

Figure 1.1

The computer-related number systems.

DECIMAL (10)	BINARY (2)	HEXIDECIMAL (16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10



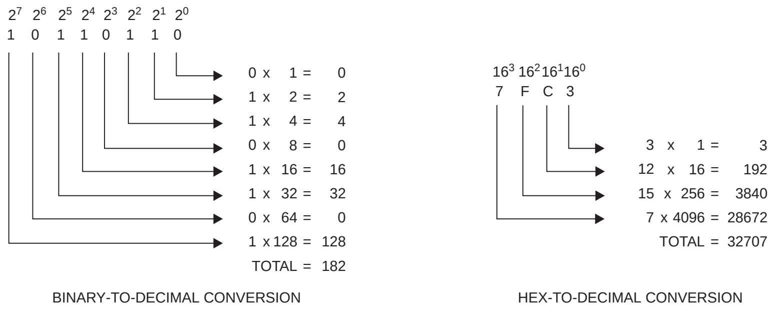
A binary number (10011101) is shown as it would be stored in an 8-bit register. When the byte is divided into its two nibbles, the number can be represented more conveniently in terms of its hex equivalent 9D. When expressing hexadecimal values, it is common to follow the value with a lower case h (for example, 9Dh).

Converting between binary and hexadecimal is a simple matter of grouping the binary word into its respective nibbles and then converting each nibble into its hex equivalent. However, converting between either of these systems and decimal (base-10) is another matter all together.

In order to convert from either of these systems to decimal, each bit, or digit, must be multiplied by its weighted positional value, as depicted in Figure 1.2.

Figure 1.2

Converting other number systems to decimal.



Due to the complexity of this operation, computer words are rarely converted into decimal unless done so by the computer's hardware. Whichever number system is being used, it should be remembered that it's being used only for the convenience of humans, because all the computer can understand is the high and low logic level voltages used to represent binary numbers.

Numeric Data Words

Objectives

Now that the discussion of computer word length and representation is complete, it is appropriate to return to the topic of the types of words used with computers.

The first type of word discussed is the pure `numeric data word`. As the name implies, these words contain only numeric information and represent a quantity in binary, hexadecimal, or a specialized form of binary called `binary coded decimal` (BCD). In the BCD form of binary, the count pattern is limited to binary representation between 0 and 9. After the binary count reaches 9, the next count clears the first BCD digit to 0, and advances the second BCD digit to 1, creating a binary equivalent of the decimal number system.

Like hexadecimal, the BCD system groups binary digits in 4-bit digits. But unlike hexadecimal, BCD does not use the binary codes for numbers 10 through 15. These codes are not used in the BCD system and would be considered invalid if they were encountered in a system using BCD. Also, unlike hex, the BCD system can be converted quite easily to decimal because it is not a true weighted number system. Figure 1.3 illustrates the usage of the BCD number system.

Figure 1.3

The BCD code.

BASE 10	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000

9	4	3	=	943_{10}
1001	0100	0011	=	BCD EQUIVALENT

Another method of representing binary values is the `octal,` or `base-8,` numbering system. Although not as common as hexadecimal, some computers group binary numbers into a 3-bit pattern as depicted in Figure 1.4. The highest binary number that can be represented using this number system is a “7.”

Figure 1.4

The octal code.

OCTAL (8)	BINARY (2)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111
10	001 000

6	7	2	4	=	6724_8
110	111	010	100	=	OCTAL EQUIVALENT

Alphanumeric Words

Objectives

In addition to manipulating numeric values, the computer must also be capable of handling `alphanumeric data` (letters and special characters). Once again, coded binary bit patterns are used to represent these characters and symbols. When you type an alphabetic character on your keyboard, what the key actually does is generate a specific binary bit pattern that the computer recognizes as that particular character. In order to display the character on your monitor, the output device must reconstruct it from the bit pattern stored in the computer.

The alphanumeric code most commonly used in personal computers is the `ASCII (American Standard Code for Information Interchange)` code. In ASCII, characters of the alphabet (both upper and lower case), decimal numbers 0 to 9, and a variety of special symbols and punctuation marks are each assigned a specific 7-bit binary pattern. The 7-bit ASCII code is depicted in Figure 1.5.

Figure 1.5

ASCII Code.

CTRL	CHARACTER	BINARY BIT 7 TO BIT 0	OCTAL	DECIMAL	HEXADECIMAL	CHARACTER	BINARY BIT 7 TO BIT 0	OCTAL	DECIMAL	HEXADECIMAL	CHARACTER	BINARY BIT 7 TO BIT 0	OCTAL	DECIMAL	HEXADECIMAL
@	NUL	00000000	000	000	00	+	00101011	053	043	2B	V	01010110	126	086	56
A	SOW	00000001	001	001	01	,	00101100	054	044	2C	W	01010111	127	087	57
8	STX	00000010	002	002	02	-	00101101	055	045	2D	X	01011000	130	088	58
C	ETX	00000011	003	003	03	.	00101110	056	046	2E	Y	01011001	131	089	59
D	EOT	00000100	004	004	04	/	00101111	057	047	2F	Z	01011010	132	090	5A
E	ENQ	00000101	005	005	05	0	00110000	060	048	30	[	01011011	133	091	5B
F	ACK	00000110	006	006	06	1	00110001	061	049	31	\	01011011	134	092	5C
G	BEL	00000111	007	007	07	2	00110010	062	050	32	]	01011101	135	093	5D
H	BS	00001000	010	008	08	3	00110011	063	051	33	^	01011110	136	094	5E
I	MT	00001001	011	009	09	4	00110100	064	052	34	-	01011111	137	095	5F
J	LF	00001010	012	010	0A	5	00110101	065	053	35	'	01100000	140	096	60
K	VT	00001011	013	011	0B	6	00110110	066	054	36	a	01100001	141	097	61
L	FF	00001100	014	012	0C	7	00110111	067	055	37	b	01100010	142	098	62
M	CR	00001101	015	013	0D	8	00111000	070	056	38	c	01100011	143	099	63
N	SO	00001110	016	014	0E	9	00111001	071	057	39	d	01100100	144	100	64
O	SI	00001111	017	015	0F	:	00111010	072	058	3S	e	01100101	145	101	65
P	DLE	00010000	020	016	10	;	00111011	073	059	3B	f	01100110	146	102	66
Q	DC1	00010001	021	017	11	<	00111100	074	060	3C	g	01100111	147	103	67
R	DC2	00010010	022	018	12	=	00111101	075	061	3D	h	01101000	150	104	68
S	DC3	00010011	023	019	13	>	00111110	076	062	3E	i	01101001	151	105	69
T	DC4	00010100	024	020	14	?	00111111	077	063	3F	j	01101010	152	106	6A
U	NAK	00010101	025	021	15	@	01000000	100	064	40	k	01101011	153	107	6B
V	SYN	00010110	026	022	16	A	01000001	101	065	41	l	01101100	154	108	6C
W	ETB	00010111	027	023	17	B	01000010	102	066	42	m	01101101	155	109	6D
X	CAN	00011000	030	024	18	C	01000011	103	067	43	n	01101110	156	110	6E
Y	EM	00011001	031	025	19	D	01000100	104	068	44	o	01101111	157	111	6F
Z	SUB	00011010	032	026	1A	E	01000101	105	069	45	p	01110000	160	112	70
[	ESC	00011011	033	027	1B	F	01000110	106	070	46	q	01110001	161	113	71
\	FS	00011100	034	028	1C	G	01000111	107	071	47	r	01110010	162	114	72
]	GS	00011101	035	029	1D	H	01001000	110	072	48	s	01110011	163	115	73
^	RS	00011110	036	030	1E	I	01001001	111	073	49	t	01110100	164	116	74
_	US	00011111	037	031	1F	J	01001010	112	074	4A	u	01110101	165	117	75
	SP	00100000	040	032	20	K	01001011	113	075	4B	v	01110110	166	118	76
	!	00100001	041	033	21	L	01001100	114	076	4C	w	01110111	167	119	77
	"	00100010	042	034	22	M	01001101	115	077	4D	x	01111000	170	120	78
	#	00100011	043	035	23	N	01001110	116	078	4E	y	01111001	171	121	79
	$	00100100	044	036	24	O	01001111	117	079	4F	z	01111010	172	122	7A
	%	00100101	045	037	25	P	01010000	120	080	50	{	01111011	173	123	7B
	&	00100110	046	038	26	Q	01010001	121	081	51	|	01111100	174	124	7C
	'	00100111	047	039	27	R	01010010	122	082	52	}	01111101	175	125	7D
	(	00101000	050	040	28	S	01010011	123	083	53	~	01111110	176	126	7E
	)	00101001	051	041	29	T	01010100	124	084	54	DEL	01111111	177	127	7F
	*	00101010	052	042	2A	U	01010101	125	085	55					

CTRL	ABBR.	DESCRIPTION	CTRL	ABBR.	DESCRIPTION	CTRL	ABBR.	DESCRIPTION
@	NUL	null,or all zeros	K	VT	vertical tabulation	V	SYN	synchronous idle
A	SOH	start of the heading	L	FF	form feed	W	ETB	end of transmission block
B	STX	start of text	M	CR	carriage return	X	CAN	cancel
C	ETX	end of text	N	SO	shift out	Y	EM	end of medium
D	EOX	end of transmission	O	SI	shift in	Z	SUB	substitute
E	ENQ	enquiry	P	DLE	datalink escape	[	ESC	escape
F	ACK	acknowledge	Q	DC1	device control 1 (X ON)	\	FS	files separator
G	BEL	bell	R	DC2	device control 2	]	GS	group separator
H	BS	backspace	S	DC3	device control 3 (X OFF)	^	RS	record separator
I	HT	horizontal tabulation	T	DC4	device control 4	_	US	unit separator
J	LF	line feed	U	NAK	negative acknowledge		SP	space
							DEL	delete

Although the basic ASCII code contains only seven bits, it's an 8-bit-oriented world inside the computer. For some applications, a "0" may be added to the MSB of all the ASCII bit patterns, or a "1" may be used to produce an extended (non-standard) ASCII code of 256 characters. But more often, the extra bit is used as an error-checking bit when ASCII is used for the transmission of alphanumeric data between the computer and its peripherals or another computer.

Under these circumstances, the extra bit, now called a `parity bit`, assumes a certain value dependent upon the number of 1s in the character. This bit is used to check the integrity of the transmitted character. Parity checking can be implemented in two forms:

- `Odd parity`—where the value of the parity bit assumes a value of "1" or "0" so that the total number of "1" bits in the character is odd.
- `Even parity`—where the parity bit assumes a value of "1" or "0" to make the total number of "1" bits in the character equal to an even number.

When ASCII is being used to transfer information between the computer and one of its peripherals (such as the keyboard, the monitor, or a disk drive), then circuitry in the computer or the peripheral generates the proper type of parity bit as the ASCII characters are transmitted. On the receiving end, the parity bit is regenerated from the received data and compared to the parity bit that was transmitted with the character. If the two bits agree then the communicating devices accept the transmission as a good one and continue. It should be noted that parity checking is the simplest form of error checking used in computer systems to guarantee the integrity of transmitted data. However, parity checking is capable of detecting only single-bit errors. More advanced error checking and correction schemes are widely used to check for and correct data transmission errors.

Note

> Notice in the ASCII code table that some codes are specified as *control codes*. These characters are special codes that can be used to modify functions associated with other characters. They are also frequently used when working with peripheral devices to specify certain operations or conditions that need to be performed or attended to.

Instruction Words

Objectives

The last type of computer word to be discussed is the `instruction word`. Unlike binary and coded data words, an instruction word is not used to convey any kind of numeric value or character reference. Instead, this kind of word tells the computer what operation to perform and where to find the data (if any) on which to perform the operation. No data is contained in the word, just a complete instruction to the computer. Although the other two types of words are common to most computers, each computer has its own set of operations that it can perform under the direction of a set of instruction words. From computer to computer, the format of instruction words can vary greatly. But generally, instruction words for most computers convey the same types of information.

Figure 1.6 depicts three possible instruction word formats that could be used with an 8-bit computer, depending on the type of operation being specified. In the first example, a `1-byte instruction` containing only coded information about what operation is to be performed (operation code, or `op code`) is presented. This type of instruction is used in operations such as a HALT instruction, where no data is required to carry out the instruction. The second example depicts a `2-byte instruction`. Here, the first byte once again contains the op code and the second byte contains the operand (a piece of data to be worked on) address. The last example illustrates a `3-byte instruction`. The op code is followed by two pieces of data. This format can also be used to indicate a particular operation to be performed, the location of a data word, and a location to store the result of the operation.

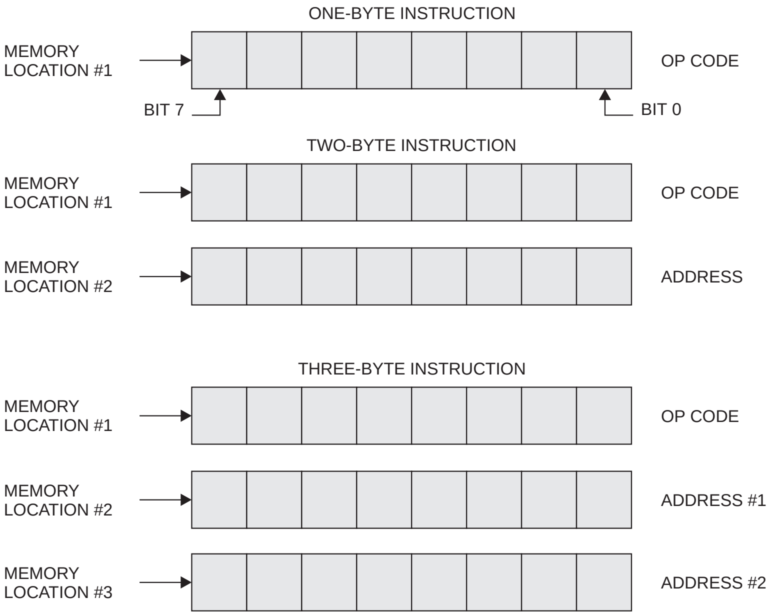


Figure 1.6

Instruction word formats.

These instruction words should not be confused with the instructions given in a higher order language, such as BASIC. The instruction formats described above pertain to the `machine language`, 1s and 0s understood by a computer's internal hardware. A single BASIC language instruction can generate several lines of machine language instructions after it has been converted.

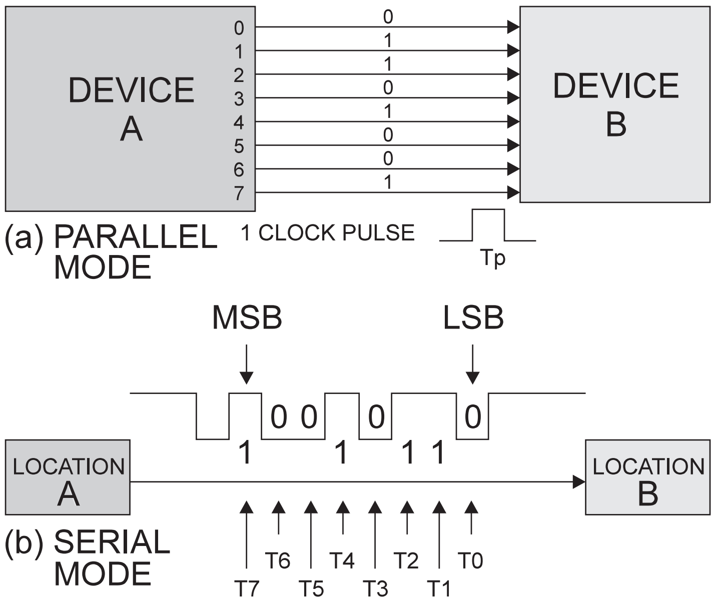
Moving Words Around

The most frequent operation performed in a computer is the movement of information from one location to another. This information is moved in the form of `words`. Basically, the words can be transferred in two modes. These modes are `parallel mode`, where an entire word is transferred from location A to location B by a set of parallel conductors at one instant, and `serial mode`, where the bits of the word are transmitted along a single conductor, one bit at a time. Serial transfers require more time to accomplish than parallel transfers because a clock cycle must be used for each bit transferred.

A parallel transfer requires only a single clock pulse. An example of both parallel and serial transfers is depicted in Figure 1.7. Because speed is normally of the utmost importance in computer operations, all data movements within the computer are conducted in parallel, as shown in . But when information is being transferred between the computer and its peripherals (or another computer), conditions may dictate that the transfer be carried out in serial mode, as shown in . Parallel and serial transmissions, as well as the conversions that must take place to convert from one mode to the other, are discussed in greater detail later in this chapter and throughout this text.

Figure 1.7

Parallel and serial data transfers.



Basic Hardware Structures

Objectives

Most of the topics discussed so far have dealt with the non-physical aspects of the computer's organization—its intelligence, if you will. The bits, bytes, words, and programs that make the computer function are referred to as `software`. The term "software" is actually used when referring to any aspect of the computer or its operation that you can't reach out and touch.

The other aspects of the computer system, such as its circuit boards, cables, connectors, magnetic disks, and so forth, are referred to as `hardware`. The term “hardware” is used to indicate any part of the computer system that you can touch with your hand.

The overall performance of any computer system is based on the capabilities of both its hardware and the software used to guide its operation. The most sophisticated computer hardware in the world is useless junk without proper software. And, conversely, the most well-written software is totally wasted if the hardware doesn’t have the capabilities to perform the operations called for in its programs.

Therefore, a digital computer can be defined as a collection of digital devices that can perform logical and mathematical operations under the direction of a program.

Integrated Circuits

The first digital computers were giants that took up entire rooms and required several technicians and engineers to operate. They were constructed with vacuum tubes and their computing power was very limited by comparison to modern computers. However, the advent of `integrated circuit` (IC) technology in 1964 launched a new era in compact electronic packaging. The much smaller, low-power transistor replaced the vacuum tube and the size of the computer began to shrink.

Very quickly, engineers realized that several transistors could be created simultaneously on the same piece of silicon. Soon, entire electronic circuits were being integrated onto a single silicon chip (in other words, an integrated circuit).

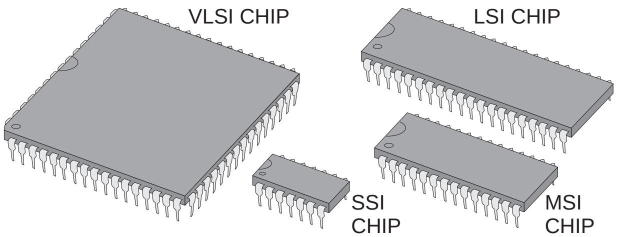
The first ICs were relatively small devices that performed simple digital logic. These basic digital devices still exist and occupy a class of ICs referred to as `Small Scale Integration` (SSI) devices. SSI devices range up to 100 transistors per chip. As manufacturers improved techniques for creating ICs, the number of transistors on a chip grew and complex digital circuits were fabricated together. These devices are categorized as Medium Scale

Integration (MSI) devices. MSI devices range between 100 and 3,000 circuit elements. Eventually, `Large Scale Integration` (LSI) and `Very Large Scale Integration` (VLSI) devices were produced. LSI devices contain between 3,000 and 100,000 electronic components while VLSI devices exceed 100,000 elements.

IC technology today enables millions of circuit elements to be constructed on a single small piece of silicon. Some VLSI devices contain complete computer modules. These devices are commonly referred to as `Application Specific Integrated Circuits`, or `ASICs`. By connecting a few ASIC devices together on a printed circuit board, computers that once inhabited an entire room have shrunk to fit on the top of an ordinary work desk, and now, into the palm of the hand. Various integrated circuit package types are depicted in Figure 1.8.

Figure 1.8

Integrated Circuit packages.



Today, digital computers can be grouped into three general categories. These are, in order of their computing power and complexity:

- Mainframes
- Minicomputers
- Microcomputers

`Mainframes` are the largest class of computer, used to service thousands of on-line users. Mainframes can contain hundreds of megabytes of primary memory and hundreds of gigabytes of secondary memory. `Minicomputers` are medium-sized computers capable of serving several hundreds of users. `Microcomputers` are the smallest computers and typically handle a very limited number of users.

A personal computer, or PC, is a type of microcomputer that is intended for use by an individual (thus, "personal") either at home or in the work place.

Basic Computer Structure

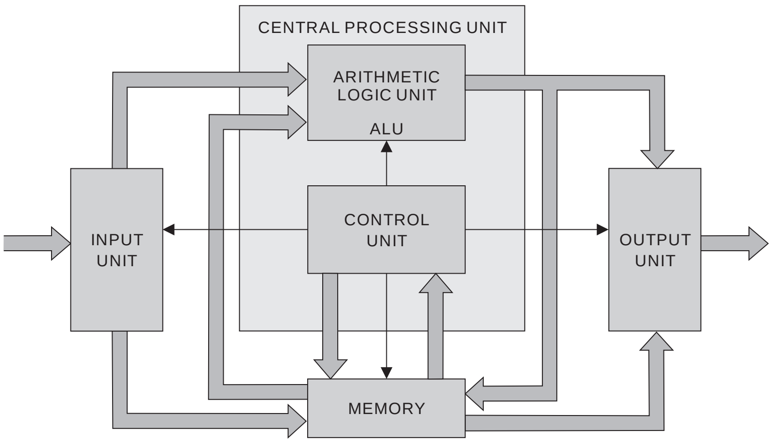
Objectives

In their most basic form, all digital computers consist of the four fundamental blocks depicted in Figure 1.9. The computer must have some type of the following:

- Central Processing Unit (CPU)
- Memory Unit
- Input Unit
- Output Unit

Figure 1.9

The fundamental blocks of a digital computer.



In many instances, an input and an output unit are combined to service some particular device or an add-on to the computer. In these cases, the combined unit is referred to as an Input/Output (I/O) unit. Three communications paths called buses normally interconnect the computer's basic units.

Central Processing Unit

The CPU section consists of two major sub-sections: an arithmetic logic unit (ALU) and a control unit. The ALU is the section of the computer where the actual math and logic operations are performed under the direction of the control unit.

The control unit receives instructions from the program, decodes them and then generates signals to inform the ALU as to what operation has been requested. In addition, the control unit orchestrates the operation of all the other logical blocks by generating the proper timing and control signals necessary to complete the execution of the instruction.

Due to the nature of its hardware structure, the typical microcomputer's operation is limited to performing one instruction at a time, involving a maximum of two quantities. The computer's true value lies in the fact that it can execute a tremendous number of instructions in a very short period of time. A typical personal computer can execute several `million instructions per second` (mips).

Microprocessors

Many times people speak of a microprocessor as the CPU. Others refer to the system's main unit as the CPU. In both cases, the terminology is faulty. In the early days of computers, the central processing unit was the area of the computer where logical and mathematical computations were carried out. It was usually composed of discrete circuits or devices.

One of the earliest LSI devices brought together a section of special on-board data storage areas, referred to as `registers`, and a CPU in a single IC package. This device is called a `microprocessor` (or MPU). In this manner, a single IC device becomes the brain of the computer.

Several microprocessor chips are on the market, each with its own unique architecture, capabilities, and instruction set (operations that it can execute). The capabilities of the microprocessor used in a particular computer ultimately determine the characteristics and capabilities of the entire computer. Basically, microprocessors are classified by the size of their internal data storage registers. Common microprocessor sizes are 8 bits, 16 bits, and 32 bits.

Basic Microprocessor Operation

The operation of all microprocessors is basically the same. They all execute programs in a cyclic manner. An `instruction cycle` is followed by an `execution cycle`. These two cycles are repeated until the program is terminated, or until it reaches its end. During the instruction cycle, the processor retrieves an instruction from memory, decodes it, and prepares to carry out the instruction as directed. The instruction may call for additional data to be retrieved from memory, manipulated in a logical manner, or to be written into a memory location. In any case, the instruction is carried out during the processor's execution cycle.

After completing the operations required by the instruction, the microprocessor moves sequentially to the next instruction. A particular instruction may cause the system to jump to another memory location to receive another instruction. This `instruction jump` can be dictated directly by the instruction, or it can be based on the outcome of some logical operation.

All microprocessors have a fixed set of operations they can perform. The operations that a given microprocessor can perform are referred to as its `instruction set`. The size of the instruction set determines how many different operations the microprocessor can perform. This, in turn, determines how quickly data can be processed.

Memory Unit

Objectives

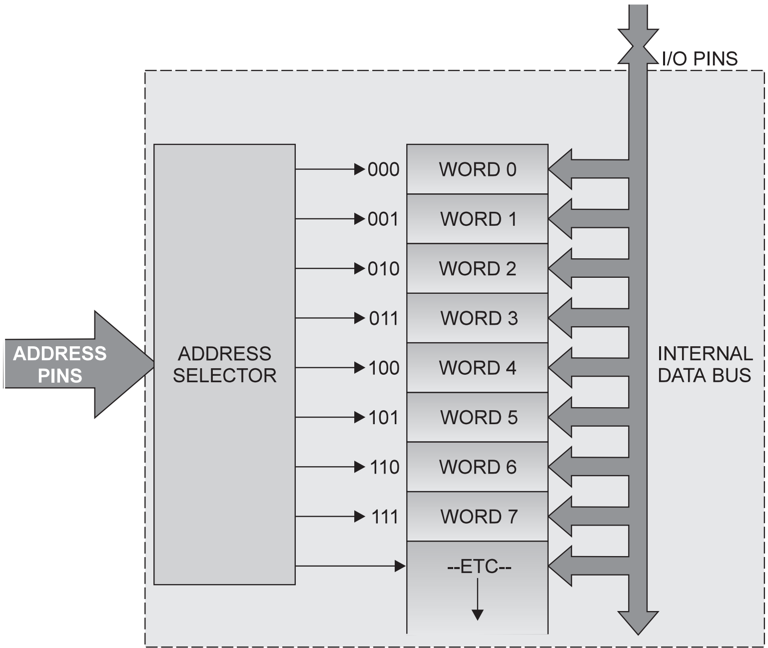
The `memory unit` is the section of the computer where instructions and data to be used by the computer are stored. The memory unit involved directly with the microprocessor consists of high-speed semiconductor devices that are compatible with the microprocessor's speed so as not to slow its operation. In times past, this type of memory was referred to simply as `internal memory`, because it was usually located in the same housing with the microprocessor. Slower, less expensive forms of memory, called `mass storage`, were located in a separate unit. Mass storage systems are used for long-term storage of programs and data, or to hold masses of programs and data too large to be held in the internal memory. In any event, the main emphasis for mass storage devices or systems is the capability to store large amounts of data on a permanent, or semi-permanent basis, as cheaply as possible.

With the advent of the microcomputer, this method of differentiating memory types blurred. In these computer models mass storage systems were incorporated into the same housing as the basic system unit. In order to make a clear distinction between the two memory types, the fast semiconductor memories used directly with the microprocessor are referred to as `primary memory,` while the slower, less expensive bulk memory systems are designated as `secondary memory`. In this section, the discussion refers to only the primary memory types.

Semiconductor memories can be thought of as a collection of systematically arranged boxes in which computer words are stored. This concept is illustrated in Figure 1.10. Notice that the boxes are arranged so that each has its own unique location, which is specified by a number. This number is referred to as the memory location's `address`. When the microprocessor wishes to access a particular box to store or retrieve information in it, it does so by generating the address of that particular storage space, along with special signals required to perform the operation.

Figure 1.10

The logical structure of an IC memory device.



A `Read` operation, also referred to as a `Fetch` operation, is one where the contents of a specific memory location are sensed by the microprocessor. On the other hand, the act of placing a new word in a specific address is called a `Write` or `Store` operation. When the new word is placed in a memory location, it replaces any data that was previously stored there.

Semiconductor memories are actually of two major types:

- Read-Only Memories (ROM)—hold data on a permanent basis.
- Random Access Memories (RAM)—are used for temporary storage of data.

Note

ROM memories generally hold data that was programmed into them at the factory and are not intended to be changed.

ROM Memory

Objectives

The description found in the preceding note is the classical definition generally applied to ROM, but in reality there are several types of ROM, some of which can be erased and reprogrammed (but not during the normal operation of the computer). These classes include the following:

- Mask-Programmed ROM (MROM)—programmed at the factory.
- Programmable ROM (PROM)—can be custom-programmed by the user (once) using special circuitry.
- Erasable-Programmable ROM (EPROM)—can also be programmed and erased by the user using ultraviolet light and special circuitry external to the computer.
- Electrically Erasable PROM (EEPROM)—can be erased and reprogrammed by special circuitry within the computer.

The one thing all forms of ROM have in common is that they are all `non-volatile`. This means that the data contained in the memory is not lost when the computer is turned off or when electrical power is lost. This enables the computer to begin reading instructions and data from this type of memory as soon as it is powered up.

The term "`Read-Only`" truly applies to MROM and PROM memories, which are written once and then cannot be erased or rewritten. The other ROM classes are more appropriately referred to as `Read Mostly Memories,` where the ratio of Read operations to Write operations is very high. The generic term "Read-Only" is used with all non-volatile, semiconductor memories that cannot be written to during the normal operation of the computer.

ROM Applications

Objectives

The different classes of ROM memories are used to perform a wide variety of applications within the computer. Some common ROM applications are listed as follows:

- Firmware storage
- ROM lookup tables
- Code converters
- character generators for printers and video displays

The most common of these applications include use as `firmware` storage. In this application, the computer's operating system programs and language interpreters are stored in ROM devices so that the computer can begin operation as soon as it's turned on.

The term "firmware" describes the fact that software is stored in hardware (in this case ICs) on a permanent basis. As an example, the computer's starting address is commonly stored in ROM. This is the address from which the microprocessor takes its first instruction after it has been turned on or reset.

ROM is also used to store tables of data, such as trigonometric functions and code conversion tables, which do not change. Rather than performing a mathematical manipulation each time a function such as sine, cosine, tangent, pi, and so forth is needed, the microprocessor simply looks up the value associated with the function in a `ROM table.` The CPU does this by decoding an address from the function command, such as sine 37 degrees, and applying it to the ROM table. The output of the table is the numerical value of the function that was stored at that address.

In like manner, ROM tables are often used to convert characters from one code to another within the computer. As an example, Figure 1.11 depicts the ASCII representation of the number 9 being applied to a ROM conversion table as an address, to which the ROM table responds by producing the binary equivalent of 9, for use by the computer.

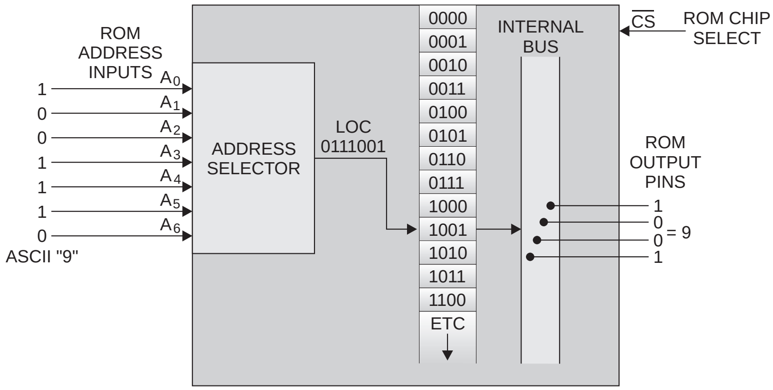


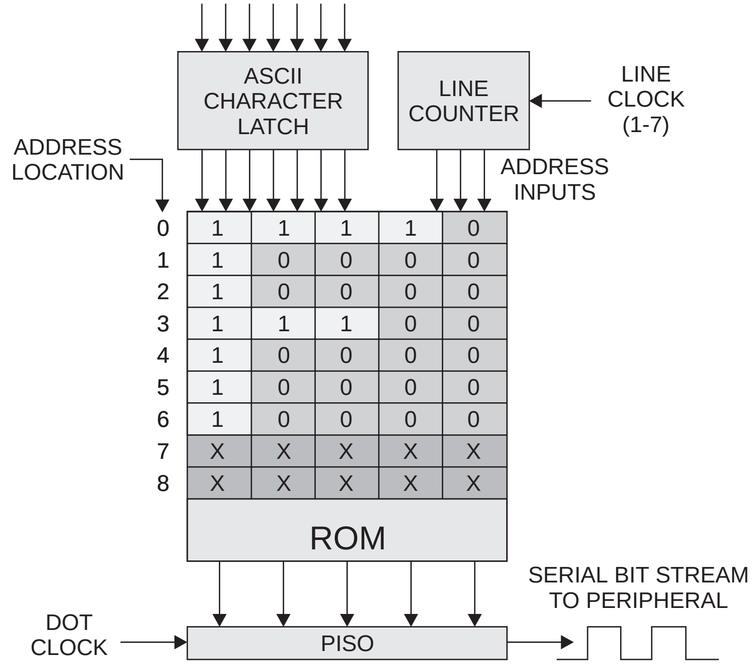
Figure 1.11

A ROM code conversion generator.

In addition, `ROM memories` are frequently used to store dot pattern codes for the character generator sections of alphanumeric output devices, such as video monitors and some types of printers. Figure 1.12 depicts the dot pattern of the letter "F," arranged in a 5×7 matrix pattern. The letter is read from the ROM by first applying the proper code (such as the ASCII code for the upper case F, in this case) as part of the character's address. In order to read the entire dot pattern of the character, a segment of the address is produced by a binary counter whose output is used to step through the dot pattern one horizontal row at a time. In this case, the counter's initial count of "0" produces the first horizontal row, and its final count of "6" produces the last, or seventh, row of dots. If a different character must be produced, the ASCII code for that character would simply be used as the upper portion of the address.

Figure 1.12

A ROM character generator.



Several standard ROM lookup tables, code conversion tables, and dot matrix character generator tables are available from IC manufacturers. By storing such information in ROM chips, computer and peripheral manufacturers build a great deal of flexibility into their systems and devices.

As shown in the preceding examples , it is possible to completely alter the manner in which a computer or peripheral goes about performing its basic function by simply exchanging a handful of IC chips. If an improved operating system is developed for a computer, or a different type style or character font is desired for a printer, the only thing that is required is to change the ROM.

In newer computer systems, special EEPROMs, referred to as `Flash ROMs`, enable the operation of the system to be re-defined through software. In these units, it is not necessary to change the ICs to upgrade the system. Simply install the new information from an acceptable source, such as a disk.

RAM Memory

Objectives

The other type of high-speed semiconductor memory used with computers and peripheral devices is IC `random access memory` or `RAM`. The term "random access" means that any address location in the memory can be accessed as quickly as any other location. Because there are other types of RAM memory, IC devices used for primary memory are more appropriately referred to as `Read/Write` (R/W) `memories`. In the case of primary memory, the generic term "RAM" always refers to semiconductor R/W memory.

Semiconductor RAM memories are fast enough to work directly with the microprocessor without slowing it down. The computer uses the RAM portion of primary memory to hold programs and data currently being executed by the microprocessor. During the execution of a program, the contents of many RAM address locations are changed as the microprocessor updates the program, by storing intermediate or final results of operations performed.

Like semiconductor ROM, semiconductor RAM has more than one type. As a matter of fact, it has two general categories:

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Although they both perform the same function, the methods they use are completely different. Static RAM stores binary bits in such a manner that the bits remain in RAM as long as power to the chip is not interrupted. Dynamic RAM, on the other hand, requires that stored data be `refreshed`, or rewritten, periodically to keep it from fading away. As a matter of fact, each bit in the DRAM must be refreshed at least once every 2 milliseconds or the data dissipates. Because it can't be assumed that each bit in the memory will be accessed during the normal operation of the system (within the time frame allotted), special circuitry is required to constantly refresh the data in the DRAM.

Although the extra circuitry and inconvenience associated with refreshing may initially make DRAM memory seem to be a distant second choice behind static RAM, this is not the case. The truth is that, due to the simplicity of DRAM's internal structure, the bit storage capacity of a DRAM chip is much greater than that of a similar static RAM chip and offers a much lower rate of power consumption. Both these factors contribute to making DRAM memory the economical choice in certain RAM memory systems, even in light of the extra circuitry necessary for refreshing.

Generally, static RAM is used in smaller memory systems where the added cost of refresh circuitry would greatly add to the cost per bit of storage. DRAM is used in larger memory systems where the extra cost of refresh circuitry is distributed over a greater number of bits and is offset by the reduced operating cost associated with DRAM chips.

Whether the RAM section of primary memory is made up of static or dynamic RAM chips, all RAM has the disadvantage of being `volatile`. This means that any data stored in RAM is lost if power to the computer is disrupted for any reason. On the other hand, both types of RAM have the advantage of being fast: They can be written into and read from with equal ease.

Buses

Objectives

All of the basic components of the computer are tied together by communications paths called `buses`. A computer bus is simply a parallel collection of conductors that carry data and control signals from one unit to another. Any computer has three major system buses identified by the type of information they carry. The three major system buses are listed as follows:

- Address bus
- Data bus
- Control bus

These buses are actually extensions of the microprocessor's internal communications structures.

Address Bus

Objectives

The address bus is a `unidirectional` pathway that carries addresses generated by the microprocessor to the memory and I/O elements of the computer. The size of the address bus, determined by the number of conductors in the bus, determines the number of memory locations and/or I/O elements the microprocessor can address. If the address bus is composed of 16 lines (or bits), the microprocessor can generate 2^{16}, or 65,536 distinct address codes. If the address bus size is increased to a 20-bit word size, the microprocessor's capability to address memory and I/O elements is increased to 2^{20}, or 1,048,576 possible addresses.

When discussing addressing capacity, it is common to use the letter "K" (for `kilo`) to represent 1024 (2^{10}) addresses. Using this terminology, the 16-bit bus example above is capable of addressing up to 64 KB of memory, while the latter example is capable of directly addressing up to 1,000 KB of memory. This is referred to as a `Megabyte` of memory and is denoted by the letter "M" (MB for megabytes). It is also common to express addresses in hexadecimal form. As a matter of fact, address locations are very rarely specified in a decimal format.

When the microprocessor wishes to access a memory location, or an input or output element, to perform a Read or Write operation, it does so by placing the appropriate address code on its address pins (A_0-A_N) and generating the proper control signals to perform the operation. Because the memory unit is normally composed of several memory chips (RAM and ROM), special decoding circuitry is required to select the proper IC and then single out the proper memory location, input or output device that the microprocessor is trying to address.

Data Bus

In contrast to the address bus, the data bus is `bi-directional` in nature. Data flows along the data bus from the microprocessor to memory during a Write operation. Conversely, data moves from memory to the microprocessor during a Read operation. The direction for data movement is the same for Read and Write operations between the microprocessor and Input/Output devices.

Because all the computer elements must share the data bus, any device connected to the bus must have the capability to put its outputs in a high impedance state (floating) when not involved in an operation with the microprocessor. This prevents data from more than one source from being placed on the bus at one time. If two devices attempt to place data on the bus at the same time, confusion and damage to the devices results. The size of the data bus usually corresponds to the word size of the computer. In general, the larger the data bus, the more powerful the system. Common data bus sizes for microcomputers are 8 bits, 16 bits, and 32 bits.

Control Bus

Objectives

The `control bus` carries the timing and control signals necessary to coordinate the activities of the entire system. Unlike the other two busses, the control bus signals are not necessarily related to each other. Some are output signals from the microprocessor, while others are input signals to the microprocessor from Input and Output elements. Each different microprocessor type has its own unique set of control signals, which it can generate or respond to. Many control bus signals are common to most microprocessors (or similar to those used by most processors). The following are the more common control signals in use today:

- System Clock (SYSCLK)
- Read/Write Line (R/W Line)
- Memory Read (MEMR)
- Memory Write (MEMW)
- I/O Read (IOR)
- I/O Write (IOW)

One of the most important control signals in any microprocessor-based system is the `system clock`. This signal provides the timing information around which all the system's activities take place. Depending upon the type of microprocessor being used, the clock signals may be generated on the microprocessor chip, or by special IC signal generators. Microprocessors with internal clock generators usually require that an external crystal, or RC network, be connected to their clock input pins.

The control bus also carries the signals that enable selected memory or I/O elements for Read and Write operations. These signals may range from a simple `Read/Write` line (R/W) to a collection of signals such as `Memory Read` (MEMR), `Memory Write` (MEMW), `I/O Read` (IOR), and `I/O Write` (IOW). These signals are used by the microprocessor in conjunction with addresses on the address bus to perform Read and Write operations at selected memory or I/O locations.

Microprocessor Operation

To better understand how a microprocessor-based system functions, consider a simplified computer system, which, for the purposes of this example, will be called the `$1.98 Computer`. This system is based on a mythical 8-bit microprocessor, which has a 4-bit address bus and is capable of performing 16 different operations. The 4-bit address bus means that this processor is capable of addressing only 16 different memory locations, but, for these applications, this should be plenty. Consider your mythical microprocessor, its internal block diagram, and a 16X8 (16 address locations, each storing 8 bits) RAM memory block. The computer's input and output units do not actually come into play during the discussions of the system's operation. Simply assume that the programs have been entered into the RAM memory through the input unit and may be displayed through the output unit.

Internal Registers

The microprocessor consists of a group of `Internal Registers`, an `Arithmetic Logic Unit` (ALU), and a `Control Unit`. Different microprocessors have different numbers and types of internal registers. The ones depicted here are the same as, or similar to, the registers found in nearly any microprocessor. They are described as follows:

Accumulator (ACCUM): This type of register is generally used by the microprocessor to store the results of ALU operations. It is also a source of one operand for most ALU operations. Many microprocessors contain more than one accumulator register.

Program Counter (PC): This register/counter keeps track of Instruction Addresses and is always pointing at the address of the next instruction to be fetched from memory. Each time an instruction is fetched from memory, the PC is incremented by 1. The control unit may cause the PC to jump to an address out of its normal sequential order. When the control unit receives a branch instruction, such as a JUMP (JMP) or a JUMP-ON-ZERO (JPZ) command, the control unit causes the PC to be loaded with an address portion of the instruction word. Upon execution of the jump instruction, the PC continues its normal sequential count, beginning at the new address.

Note

In the example, the PC is automatically reset to a value of 0. This corresponds to the beginning of the program. In real computers, the program counter is reset to some predetermined value, such as a memory location in ROM memory containing the monitor program. This location is referred to as the microprocessor's *vector address* and is determined by the manufacturer of the microprocessor.

Address Register (AR): This register is used to hold the address currently being accessed by the microprocessor. The AR can be loaded from two different places, depending upon which part of the computer cycle is in progress. During an instruction cycle, the contents of the PC is loaded into the AR. Throughout the execution cycle, the AR is used to hold the addresses specified by the operand address portion of the instruction word.

Data Register (DR): This register is used by the microprocessor's accumulator as a temporary storage place for data. Its contents can be applied to the ALU by the accumulator. The number of temporary data registers varies from one microprocessor to the next. Different microprocessors may have several of these registers, or none at all.

Instruction Register (IR): This register is loaded with the opcode portion of the instruction word during the instruction cycle, and holds it until the completion of the execution cycle.

Instruction Decoder (ID): This device receives the opcode from the IR and decodes it for the control unit.

Arithmetic Logic Unit (ALU): The ALU performs math and logic operations under the direction of the control unit.

Control Unit (CU): The control unit is responsible for generating all the timing and control signals required for the system to execute the instructions contained in the program.

Flag Register: The flag register is not exactly a register in the classical sense. Instead, it is a collection of unrelated bits used to indicate the status of different microprocessor conditions. In the example, the Z-bit of the flag register is set if the last ALU operation produced a result of zero. Likewise, the C-flag is set if the preceding operation produced a carry bit beyond the MSB of the accumulator register. Different microprocessors have different numbers of flags in their flag registers. The microprocessor uses these flag bits to enable conditional branching to occur during the execution of the program, with the decision to branch depending upon some condition within the microprocessor.

Miscellaneous Registers: Real microprocessors contain a number of specialized registers not covered here. Among these are:

- Index Registers, which are used by the programmer to establish and maintain tables and arrays.
- Stack Pointer Registers, which are special address registers. These registers are used to create a special area in RAM memory called the `stack`. The stack is normally

dedicated to storing the contents of the other microprocessor registers during operations such as interrupt routines.

Instruction Sets

Objectives

All microprocessors have a specific set of operations that they can perform. These operations are referred to as the microprocessor's `Instruction Set`. The instruction set defines the operation of the computer very specifically. The instruction set for the $1.98 computer is presented in Table 1.1. Because the $1.98 microprocessor requires only a 4-bit opcode and a 4-bit address code, the instruction-word format is somewhat different than the more realistic formats described in the preceding section. Instead, these instruction words use the 4 MSBs for the opcode and the 4 LSBs for the operand address.

Table 1.1

Instruction Set for $1.98 Computer.

Assembly Language (Mnemonic)	Machine Language Hex Binary	Operation Description
LDA	A 1010	Transfer the contents of the memory location specified by the operand address to the accumulator register.
ADD	B 1011	Add the contents of the memory location specified by the operand address to the contents of the accumulator, and store the results in the accumulator register.
SUB	C 1100	Subtract the contents of the memory location specified by the operand address from the contents of the accumulator, and store the results in the accumulator register.

continues

Table 1.1 Continued

Assembly Language (Mnemonic)	Machine Language Hex Binary	Operation Description
JMP	2 0010	Jump unconditionally to the address specified by the operand address. After the jump has been executed, the instructions are taken in order from the new address.
JPZ	3 0010	Jump to the address called for by the operand address, but only if the ZERO flag is set.
STA	4 0011	Store the contents of the accumulator in the memory location specified by the operand address.
STP	5 0100	STOP; halt all operations.
CMP	6 0101	Compare the contents of the memory location specified by the operand address to the contents of the accumulator. If the two are equal, the E-flag is set.
JPE	7 0111	Jump to the address specified by the operand address, if the E-flag was set by the preceding operation.

Program Execution

Objectives

Table 1.2 uses the instruction set in Table 1.1 to implement a sample hexidecimal program.

Table 1.2

Sample program for $1.98 Computer.

Memory Location	Location Contents	Operation Description
0	A8	LDA with contents of location 8

Memory Location	Location Contents	Operation Description
1	BA	ADD contents of location A to value in ACCUM and Store result in the accumulator register
2	C9	SUB contents of location 9 from value in ACCUM and Store result in the accumulator register
3	3D	JUMP to location D if ACCUM=0
4	6B	CMP the contents of location B to the value in the ACCUM, set E-flag if equal
5	7D	JUMP to location D if the E-flag if set
6	CA	SUB contents of location A from value in accumulator and store in the accumulator register
7	2D	JUMP to location D unconditionally
8	0A	DATA
9	03	DATA
A	05	DATA
B	09	DATA
C	00	DATA
D	4F	STA the contents of the ACCUM in location F
E	50	STP all operation
F	00	DATA

If you work through the program in sequence and follow the computer's instructions according to their definitions, you can see that this program performs several math functions and makes decisions based upon the information available to it. Basically, this is what every computer does. At the end of the program, you should finish with a binary 7 stored in memory location F. If not, go back through the program and follow the instructions in the order that the program dictates, performing all the instructions explicitly.

After the program has been loaded into memory, the computer operator must initiate the execution of the program by giving the computer a RUN command or signal. This signal, in turn, applies a RESET input to the microprocessor, which clears its internal registers and sets the program counter to its vector address (`0`). The RUN signal also causes the computer to enter an `Instruction Cycle`. During the instruction cycle, the following events occur:

1. The contents of the PC are loaded into the AR and placed on the address bus by the control unit, along with a Read signal on the R/W line of the control bus. Together, these two pieces of information cause the address to be accessed (in the example, location `0`) and its contents to be placed on the data bus.

2. The data on the data bus (A8 in this case) is loaded into the microprocessor. The opcode portion of the instruction word is loaded into the IR and the operand address portion (8) is placed in the AR, replacing the preceding address. Both these operations are performed by the control unit.

3. The IR applies the opcode to the ID, which decodes it for the control unit. The AR places the operand address on the address bus, and the control unit increments the program counter by 1 (to memory location 1).

When the program counter is advanced to the next instruction address, the instruction cycle is ended and the computer enters an `Execution Cycle`. During this time, the instruction called for is carried out. At the end of the execution cycle, the computer automatically enters another instruction cycle, where the same sequence of events is repeated. The computer continues to perform instruction cycles, followed by execution cycles, until it receives a STOP instruction from the program.

At the beginning of the execution cycle, the control unit issues a Read signal on the R/W line of the control bus, and the data word in the memory location is placed on the data bus. In this instance, the instruction calls for the data to be loaded into the microprocessor's accumulator register. During the execution cycle, the control unit develops the signals necessary to latch the data in the accumulator.

Three other courses of action are possible during the execution cycle. They are listed as follows:

> Scenario 1: In the event that the instruction requires an ALU operation, the data word is transferred to the ALU, where it may be placed in the accumulator or the data register. The control unit must generate the control signals required to produce the transfer, and also those necessary to carry out the ALU operation.
>
> Scenario 2: If the instruction calls for data to be placed in memory, a `store` operation, the control unit first moves the data to the data register and then places it on the system's data bus. The control unit also generates a Write signal on the R/W line, which causes the data to be written into the memory at the address specified by the operand address. Recall that the operand address is still being held in the AR.
>
> Scenario 3: The instruction word calls for some type of jump to occur. If an unconditional jump instruction is received, the control unit simply causes the operand address to be loaded directly into the PC register. If a conditional jump is received and the condition proves false, the execution cycle ends, and the next instruction is taken in order.

Although the $1.98 Computer demonstrates how a typical microprocessor carries out instructions and manipulates data, it does not show the complete scope of microprocessor operations. The $1.98 Computer does not provide any method of entering new instructions or data into the system. It also lacks any provisions for outputting data that has been processed. Without these capabilities, the usefulness of a microprocessor is somewhat limited. In the following section, common methods and equipment for acquiring and outputting data are investigated.

Input/Output (I/O)

Objectives

In addition to the millions of possible memory locations in a PC, thousands of addresses are typically set aside for Input and Output devices in a system. External input and output devices, also called `peripherals`, connect to the computer's bus systems through different types of interfacing circuits. The interfacing circuits make the peripherals compatible with the system. The system's microprocessor differentiates between memory and I/O addresses through the use of separate Read and Write signals as described in the preceding section.

Interface circuits are necessary because the characteristics of most peripherals differ greatly from those of the basic computer. The microcomputer is a completely solid-state digital electronic device that uses parallel words of a given length and adheres to basic digital logic levels. However, computer peripherals generally tend to be more mechanical and analog in nature.

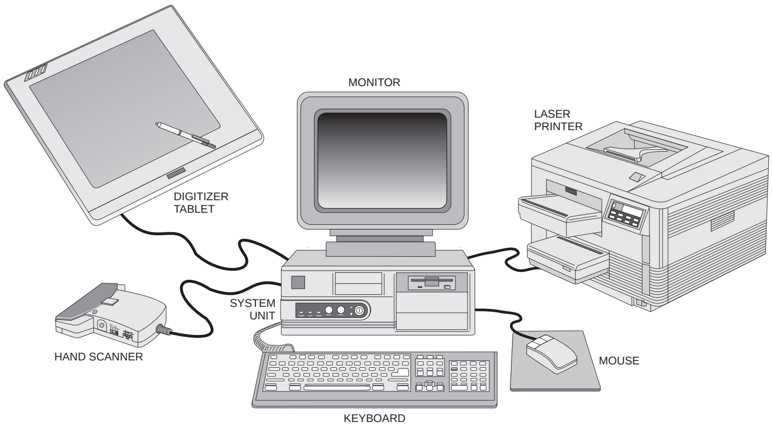
Peripherals may also use parallel or serial transmission modes between themselves and the system board. Although either transmission form may be used with any given type of peripheral, parallel buses are generally used for high-speed devices, such as disk drives and some printers. Conversely, serial transmission is used with remotely located devices or with devices whose operation is more compatible with serial data flow, such as monitors, modems, certain input devices, and some printers.

In addition to these differences, it is not uncommon for a microcomputer system to be composed of a brand-X system unit, a brand-Y printer, and a brand-Z disk drive (not to mention a host of other options from different manufacturers). This introduces a completely new set of obstacles to the orderly flow of information between the peripherals and the computer. Different manufacturers—or even a single manufacturer from one model to the next—may incorporate a wide variety of signal levels, timing, and formats into their devices which must be matched to those of the host computer. Fortunately, computer and peripheral manufacturers generally adhere to certain conventions—more or less—that enable computers to interface with a variety of different peripheral devices.

More importantly, human beings tend to be `analog` in nature. The computer's input and output units enable it to communicate with the outside world. The input units contain all the circuitry necessary to accept data and programs from peripheral input devices such as keyboards, light pens, mice, joysticks, and so forth, and convert the information into a form that is usable by the microprocessor. The input unit may be used to enter programs and data into the memory unit before execution, or it may be used to enter data directly to the microprocessor during execution. The output units contain all the circuitry necessary to transform data from the computer's language into a form that is more convenient for the `outside` world. Most often that is in the form of alphanumeric characters, which are convenient for humans to use. Common output devices include video display monitors, audio speakers, and character printers. Figure 1.13 depicts several common I/O devices associated with personal computers.

Figure 1.13

Common I/O devices used with PCs.



Some computer peripherals do double duty as both input and output units. These devices are collectively referred to as `I/O devices` and include secondary storage devices such as hard disk drives, floppy disk drives and magnetic tape drives, as well as communication devices called `modems` (modulator/demodulator). Modems enable one computer to converse with another computer over either standard or dedicated telephone lines. In the case of I/O devices, the form that data takes is not for the convenience of human beings, but instead it takes the form most suitable to carry out the function of the device.

Initiating I/O Transfers

Objectives

During a program's execution, the microprocessor constantly Reads from or Writes to memory locations. The program may also call on the microprocessor to Read from or Write to one of the system's I/O devices. Regardless of how the peripheral is connected to the system (serial or parallel), one of four methods may be used to initiate data transfer between the system and the peripheral. These four methods are listed as follows:

- `Polling:` The microprocessor examines the status of the peripheral under program control.
- `Programmed I/O:` The microprocessor alerts the designated peripheral by applying its address to the system's address bus.
- `Interrupt-driven I/O:` The peripheral alerts the microprocessor that it is ready to transfer data.
- `DMA:` The intelligent peripheral assumes control of the system's buses to conduct direct transfers with primary memory.

Polling & Programmed I/O

Both `polling` and `programmed I/O` represent software approaches to data transfer while `interrupt-driven` and `DMA` transfers are basically hardware approaches.

In the `polling method`, the software periodically checks with the system's I/O devices by testing their READY lines. When the microprocessor finds a READY line that has been asserted by a peripheral device to be ready to conduct a data transfer, it begins Reading or Writing data to the corresponding I/O port. The polling method is advantageous in that it is easy to implement and reconfigure because the program controls the entire sequence of events during the transfer. However, polling is often inconvenient because the microprocessor must be totally involved in the polling routine and cannot perform other functions. A typical polling operation is depicted in Figure 1.14.

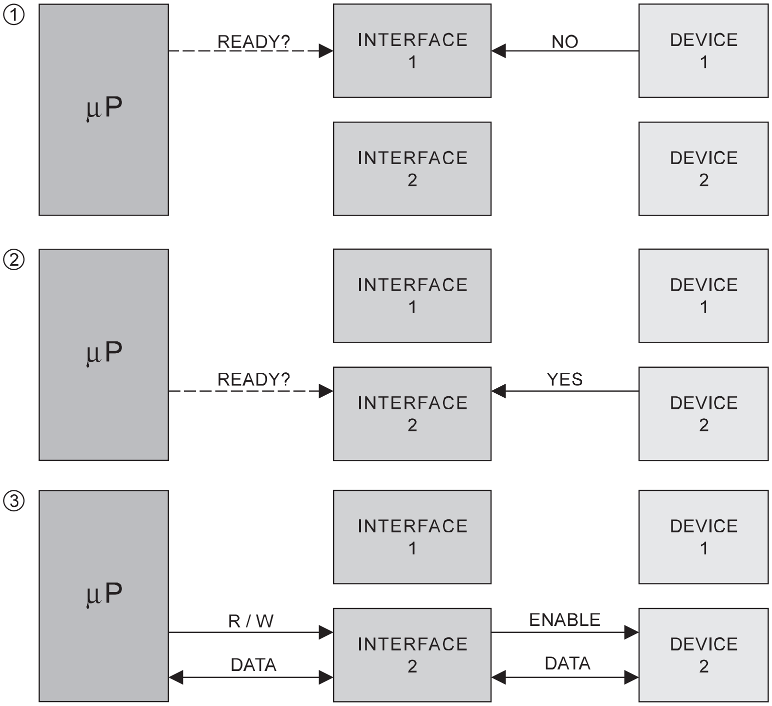


Figure 1.14

A typical polling operation.

Using the programmed I/O method to conduct data transfers calls for the microprocessor to alert the desired peripheral of an I/O operation by issuing its address to the address bus. The peripheral can delay the transfer by asserting its BUSY line. If the microprocessor receives a BUSY signal from the peripheral, it continues to perform other tasks, but periodically checks the device until the BUSY signal is replaced by a READY signal.

In order to establish an orderly flow of data during the transfer, a number of signal exchanges, or handshakes, may occur between the peripheral and the system. In a simple handshaking arrangement, the peripheral produces a byte of data at its output register when the microprocessor sends a Strobe (STB) signal to the peripheral. The microprocessor Reads the word from the data bus. The microprocessor then sends an Acknowledge (ACK) signal back to the peripheral, telling it to send the next data word. This method prevents the microprocessor from sending or requesting data at a faster rate than the peripheral can handle. In some systems, the handshaking routine is much more complex. An entire series of handshake signals may be exchanged during the transfer of a single data word. The concept of programmed I/O is illustrated in Figure 1.15.

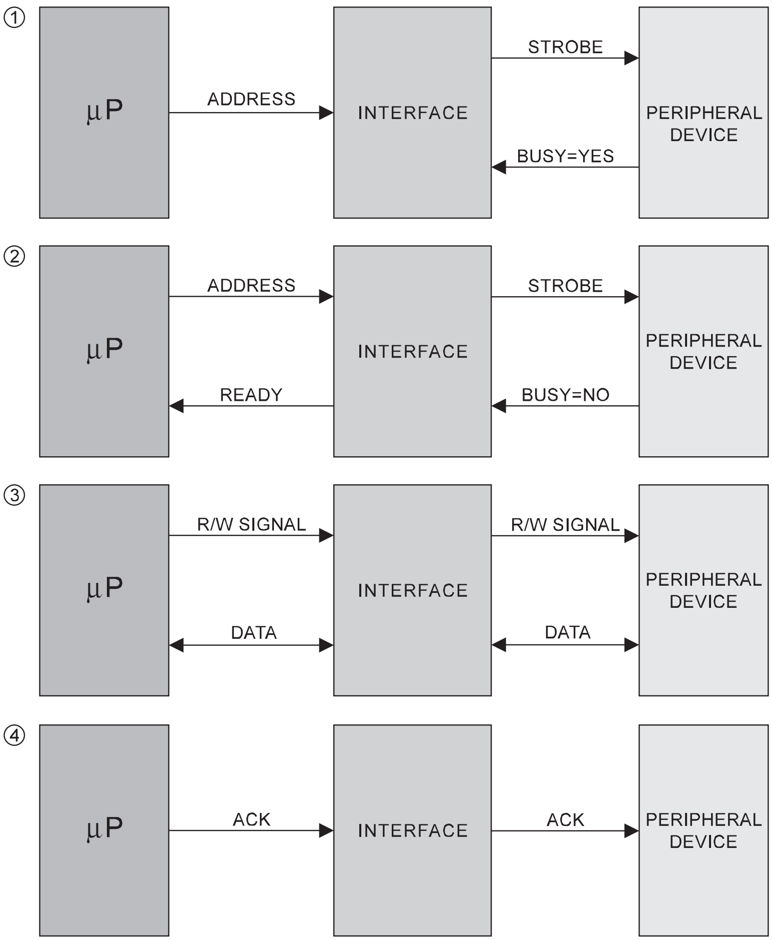


Figure 1.15

A typical Programmed I/O operation.

Interrupts

Objectives

In the course of normal operations, the various I/O devices attached to a PC, such as the keyboard and disk drives, require servicing from the system's microprocessor. Although I/O devices may be treated as memory locations, one big difference distinguishes the two; I/O devices generally have the capability to interrupt the microprocessor while it is executing a program. The I/O device does this by issuing an `Interrupt` (INT) or `Interrupt Request` (INTR) input signal to the microprocessor.

If the microprocessor is responding to INT signals and a peripheral device issues an interrupt request on an IRQ line, the microprocessor finishes executing its current instruction and issue an `Interrupt Acknowledge` (INTA) signal on the control bus. The microprocessor suspends its normal operation and stores the contents of its internal registers in a special storage area referred to as the `stack`.

The interrupting device (or an interrupt controller) responds by sending the starting address of a special program called the `interrupt service routine` to the microprocessor. The microprocessor uses the interrupt service routine to service the interrupting device. After the microprocessor finishes servicing the interrupting device, the contents of the stack are restored to their original locations, and the microprocessor returns to the original program at the point where the interrupt occurred.

Because more than one peripheral device may require the attention of the microprocessor at any given time, all computer systems have methods of handling multiple interrupts in an orderly fashion. The simplest method calls for the microprocessor, or the interrupt controller, to have multiple interrupt inputs that have a fixed priority of service. In this manner, if two interrupt signals occur at the same instant, the interrupt that has the highest priority is serviced first.

Actually, two varieties of interrupts are used in microcomputers:

- ***Maskable interrupts (MI),*** which the computer can ignore under certain conditions
- ***Non-maskable Interrupts (NMI),*** to which it must always respond

Most microprocessors have an output line called the `Interrupt Enable` (INTE), which it uses to inform peripheral devices whether it can be interrupted or not. The logic level present on this line determines whether or not the microprocessor responds to an INT or IRQ input signal. The condition of the INTE line can usually be controlled by software, which means the program can determine whether the interrupt operation is activated or not. Non-maskable interrupt inputs, on the other hand, are signals which cannot be ignored by the microprocessor and, therefore, always cause an interrupt to occur regardless of the status of the INTE line.

Think about a programmable interrupt controller IC and its relationship to the system's microprocessor. The interrupt controller chip accepts prioritized IRQ signals from up to eight peripheral devices on IRQ lines 0 through 7. When one peripheral desires to communicate with the microprocessor, it sends an IRQ to the interrupt controller. The controller responds by sending an INT signal to the microprocessor. If two interrupt requests are received at the same instance, the interrupt controller accepts the one that has the higher priority and acts on it first. The priority order is highest for the device connected to the IRQ-0 line and descends in order, with the IRQ-7 input given the lowest priority.

Direct Memory Access (DMA)

Objectives

Another difference between memory and some intelligent, high-speed I/O devices is that the I/O device may have the capability to perform data transfers (Read and Write operations) on their own. This type of operation is called `direct memory access` DMA). DMA generally involves a high-speed I/O device taking over the system's buses to perform Read and Write operations with the primary memory, without the intervention of the system microprocessor.

When the peripheral device has data ready to be transferred, it sends a `DMA request` (DREQ) signal to a special IC device called a `DMA controller`, which in turn, sends a HOLD input signal to the microprocessor. The microprocessor finishes executing the instruction on which it is currently working and places its address and data pins in a high-impedance state (floating), effectively disconnecting the microprocessor from the buses. At this time, the microprocessor issues a `buses available` (BA) or `hold acknowledge` (HLDA) signal to the DMA controller. The DMA controller, in turn, issues a `DMA acknowledge` (DACK) to the peripheral, along with the beginning address of the primary memory block to be used, and the necessary R/W and enable signals for the data transfer to begin. The key to DMA operations is that the DMA controller chip has a speed advantage over the microprocessor, in that it can transfer data bytes faster than the microprocessor can.

Actually, DMA has two distinct methods of transferring data. The crudest and simplest DMA method is referred to as HALT, or Burst Mode DMA, because the DMA controller takes control of the bus system and transfers a complete block of data to or from memory in a single burst. While the transfer is in progress, the system microprocessor sits idle, performing `No Operation` (NOP) instructions to keep its internal registers refreshed. This is the type of DMA operation performed in most computers.

The second DMA method involves the DMA controller taking control of the bus system for a shorter length of time during periods when the microprocessor is busy with internal operations and does not require access to the buses. In effect, the DMA controller steals clock cycles from the microprocessor when it's not using the bus system. This method of DMA is referred to as `cycle stealing mode`. Cycle stealing DMA is more complex to implement than Halt DMA, because the DMA controller must have the intelligence to sense the periods of time when the system buses are open.

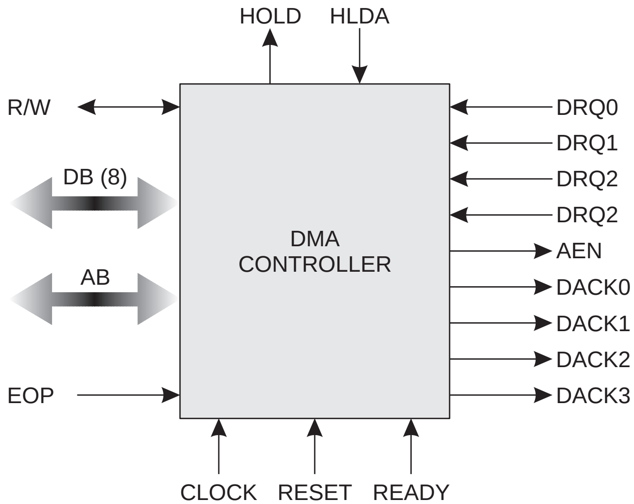
Cycle stealing DMA can be implemented as `single cycle stealing`, where the microprocessor is halted for a single clock cycle—while the DMA controller transfers a single byte—or as `full cycle stealing`, where the microprocessor is not stopped and the DMA controller can seize the buses any time the microprocessor is not using them.

Advanced microprocessors offer optimized DMA transfer capabilities because they possess the capability to load several instructions and data internally (called `queuing`) and work for an extended period of time without the need to access the bus system. The microprocessor and DMA controller can have access to the buses for varying lengths of time, as long as the DMA controller does not hold them for too many consecutive clock cycles.

Figure 1.16 depicts a typical DMA controller chip. This controller has provisions for four DMA transfer channels, with each channel consisting of a DREQ line and a corresponding DACK line. This enables the chip to conduct DMA operations for up to four devices. In addition, two of the channels may be used together to perform high-speed memory-to-memory transfers.

Figure 1.16

A typical DMA controller.



Expansion Slot Connectors

It would be very expensive to design and build a computer that fit every conceivable user application. With this in mind, computer designers include standardized connectors that enable users to configure their systems to their particular computing needs. Most personal computers use standardized `expansion slot` connectors that enable users to attach various types of peripheral devices to the system. Optional Input/Output devices, or their interface boards, may be plugged into these slots to connect the devices to the system's address, data, and control buses.

Typical options and interfaces that use these slots include the following:

- Video displays such as monitors and Liquid Crystal Display (LCD) Panels
- Hard and floppy disk drive units for mass storage

- Character/graphics printers to produce hard copy (permanent) output from the system
- Modems to enable the computer to communicate with other computers through commercial telephone lines
- Network adapters that enable computers in a local area to communicate with each other
- Game control units and other pointing devices—such as light pens and mice, and voice generation and recognition systems
- Scanners that convert images from hard copy input into digital information that the computer can process

Adapter Cards

Adapter cards plug into the expansion slots of the computer's main board, as illustrated in Figure 1.17. They typically contain the interfacing and controller circuitry for the peripheral. However, in some cases the entire peripheral may be included on the adapter card. Adapter cards enable peripheral devices to be added to the basic system to modify it for particular applications. For example, adapter cards enable less expensive devices to be used with the system for a beginner and still yet enable high-end, high-performance peripherals to be used with it for advanced applications. Several companies have developed all types of expansion cards and devices for different types of computer applications. These include I/O controllers, disk drive controllers, video controllers, modems, and proprietary Input/Output devices such as scanners.

Three important characteristics are associated with any adapter card:

- Function
- Expansion slot connector style
- Size

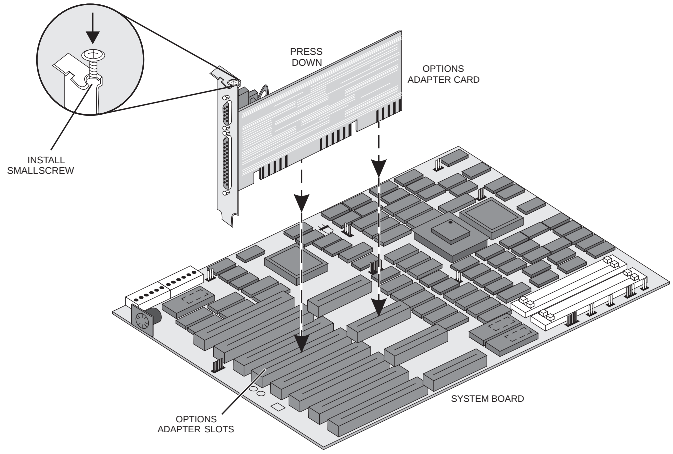


Figure 1.17

Plugging in a typical adapter card.

It is important to realize that any device connected into the system through an adapter card must have a card that is compatible with the expansion slots used in that particular type of computer.

Secondary Memory

Programs and data disappear from the system's RAM when the computer is turned off. In addition, IC RAM devices tend to be expensive, thereby making it unreasonable to construct large memories that can hold multiple programs and large amounts of data. Therefore, devices and systems that can be used for long term data storage are desirable as a second level of memory.

A number of secondary memory technologies have been developed to extend the computer's memory capabilities and store data on a more permanent basis. These systems tend to be too slow to be used directly with the computer's microprocessor. The secondary memory unit holds the information and transfers it in batches to the computer's faster internal memory when requested.

From the beginning, most secondary (also known as `bulk` or `mass`) memory systems have involved storing binary information in the form of magnetic charges on moving magnetic surfaces. This type of storage has remained popular because of three factors:

- Low cost-per-bit of storage
- Intrinsically non-volatile nature
- Successful evolution upward in capacity

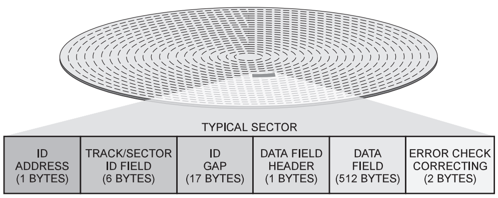
The major magnetic storage mediums are floppy disks, hard disks, and tape.

In magnetic disk systems, information is stored in concentric circles around the disk, which are referred to as `tracks`. The tracks on the disk are numbered, beginning with `0`, from the outside edge inward. The number of tracks may range from 40 up to 815, depending on the type of disk and drive being used.

Because the tracks at the outer edge of the disk are longer than those at its center, each track is divided into an equal number of equal-sized blocks called `sectors`. The number of sectors on a track may range between 8 and 50, depending upon the disk and drive type. A small hole near the center of the disk, called an `Index Hole`, marks the starting point of the #1 sector on each track. A floppy may have between 40 and 80 tracks per surface, with each track divided into between 8 and 26 sectors. Each sector holds 512 bytes. The organizational structure of a typical magnetic disk is illustrated in Figure 1.18.

Figure 1.18

The organizational structure of a magnetic disk.



In magnetic tape systems, data is stored in sequential tracks along the length of the tape, as depicted in Figure 1.19. Each track is divided into equal-sized blocks. The blocks are separated by small gaps of unrecorded space. Multiple tracks can be recorded across the width of the tape. Using multiple read/write heads, the tracks can be read simultaneously as the tape moves forward. The tracks can also be read in a serpentine manner, using a single read/write head.

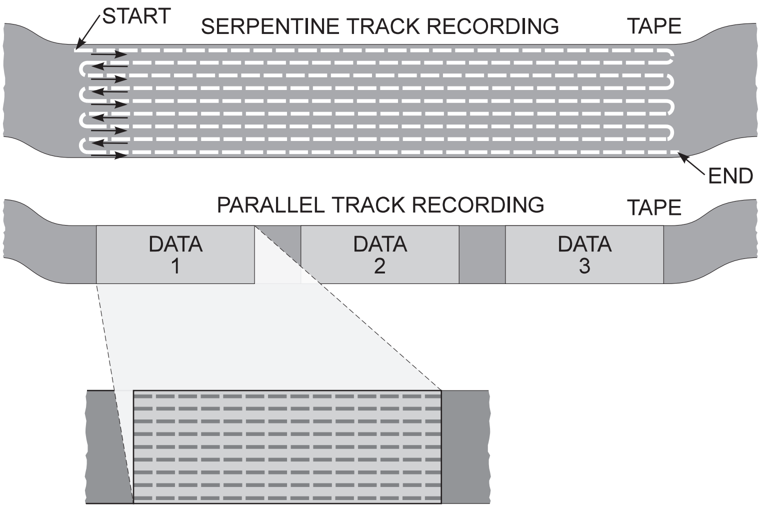


Figure 1.19

Formats for storing data on magnetic tape.

Although it is possible to directly access any sector on a magnetic disk, the sections on the tape can be accessed only in order. To access the information in block 32 of the tape, the preceding 31 blocks must pass through the drive.

Most personal computers come from the manufacturer with both a floppy-disk drive unit and a hard-disk drive unit installed. Tape generally represents a cheaper storage option, but its inherent slowness, due to its sequential nature, makes it less desirable than magnetic disks that can simply be rotated. The disks offer much quicker access to large blocks of data, at a cost that is still affordable to most users.

Floppy-Disk Drives

The most widely used data storage systems in personal computers are `Floppy Disk Drive` (FDD) units. These units store information in the form of tiny, magnetized spots on small disks that can be removed from the drive unit. After the information has been written on the disk, it remains there until the disk is magnetically erased or written over. The information remains on the disk even if it is removed from the disk drive or if power is removed from the system. Whenever the information is required by the system, it can be obtained by inserting the disk back into the drive and causing the software to retrieve (`Read`) it from the disk. Information is stored on disks in logical groupings, called `files`. A file is simply a block of related data that is grouped together, given a name, and treated as a single unit.

The disks are relatively inexpensive and are easy to transport and store. In addition, they can easily be removed and replaced if they become full.

Modern floppy disks come in 5¼ and 3½-inch diameters, like those depicted in Figure 1.20. Both types of disks are covered with a magnetic coating and encased in protective, semi-rigid envelopes. As the disks are spun inside their envelopes, the drive unit can Write data onto them from the computer's RAM memory, or Read data from them, and store it in RAM. The occurrence of these activities is signaled by a small, `Disk-Drive Activity LED` on the front of the drive unit.

The more popular of the two sizes is the 3½-inch flexible mylar disk. The actual disk is housed in a small, hard plastic envelope. The Read/Write heads access the disk surface from under a spring-loaded metal cover, which the drive unit moves out of the way. The drive spindle does not protrude through the disk. Instead, it drives the disk from a keyed metal wafer attached to the underside of the disk. A small, sliding tab in the left-front corner of the envelope performs a `Write-Protect Function` for the disk. Circuitry in the drive checks the condition of this tab to see whether it may write information onto the disk. If the tab covers the opening the disk may be written to. If, however, the opening is clear, the disk is said to be "`Write Protected`," and the drive does not write information on the disk.

A typical 5¼-inch floppy disk is also depicted in the figure. The disk's envelope has several openings. In the center of the disk is a large circular opening for the drive's spindle. The index hole is the smaller hole just beside the drive-spindle hole. It goes through both sides of the envelope and when the index hole in the disk lines up with the corresponding hole in the envelope, a light shines through the hole and is detected by circuitry on the other side of the disk. This tells the disk-drive controller that the first sector is passing under the drive's `Read/Write` (R/W) `Heads`.

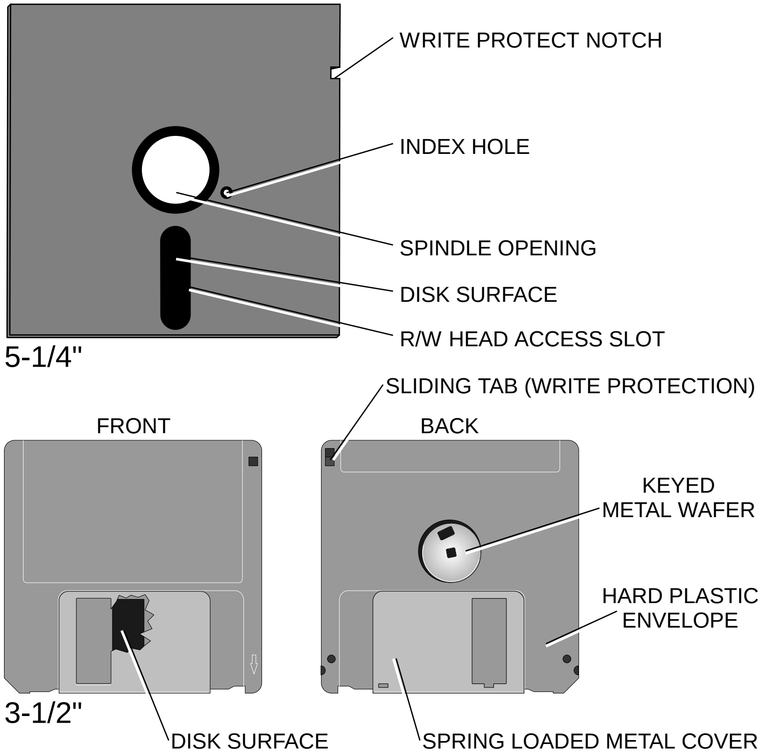


Figure 1.20

Floppy disks.

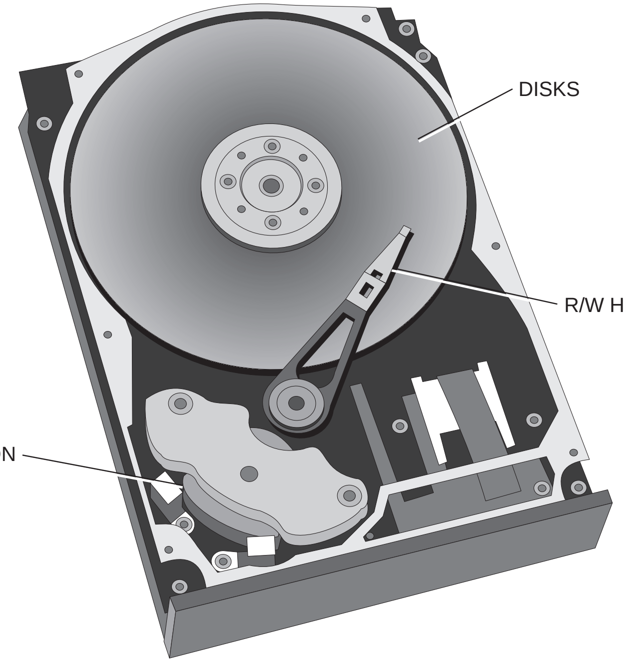
A semi-elliptical slot on each side of the envelope permits the drive's R/W heads to access the disk's surface. It is through these slots that the R/W heads Write and Read information to and from the disk. The envelope also has a small square notch along its right edge which provides the `Write Protect` function. If the notch is open the disk may be written to. Conversely, the disk cannot be written to if the notch is covered.

Hard-Disk Drives

The system's data storage potential is extended considerably by the high-speed, high-capacity `Hard-Disk Drive` (HDD) units such as the one shown in Figure 1.21. These units store much more information than floppy disks do. Modern hard drives may have storage capacities ranging up to several gigabytes. Hard drives also differ from floppy-disk units in that they use rigid disks that are permanently sealed in the drive unit (non-removable).

Figure 1.21

Inside a hard disk drive.



The disks are aluminum platters coated with a nickel-cobalt or Ferro-magnetic material. Two or more platters are usually mounted on a common spindle, with spacers between them, to enable data to be recorded on both sides of each disk. The drive's read/write mechanism is sealed inside a dust-free compartment along with the disks.

Older hard drives used disk platters that ranged in size from 8 to 40 inches in diameter. Modern hard disk drives come in sizes of 5¼-, 3½-, and 2.5-inch diameters. Of these sizes, the 5¼- and 3½-inch versions have grown in popularity due to their association with personal and business computers. However, the popularity of the 3½- and 2½-inch versions are growing with the rising popularity of smaller laptop and notebook-size computers. Hard drives ranging into the gigabytes are available for these machines.

The major differences between floppy- and hard-disk drives are storage capacity, data transfer rates, and cost. Another difference to note is the fact that hard-disk drives tend to be more delicate than floppy drives. Therefore, they require some special handling considerations to prevent both damage to the unit and a loss of data. The disks in the HDD are not removable as floppy disks are. Therefore, it is possible to fill up a hard-disk drive. When this occurs, it is necessary to delete information from the unit to make room for new information to be stored.

Conversely, floppy disks are prone to damage due to mishandling, static, temperature, etc. In addition, they are easy to misplace and they provide limited storage of application software. The latter limitation causes the user to “swap” disks in and out of floppy drives when running large programs. But, as long as you have a blank disk you do not run out of storage potential.

Peripherals

Peripherals are devices and systems that are added to the basic computer system to extend its capabilities. These devices and systems can be divided into three general categories: `Input Systems`, `Output Systems`, and `Memory Systems`. Each peripheral device interacts with the basic system through adapter boards that plug into expansion slots inside the system unit. The peripheral systems that are normally included as standard equipment in most microcomputers are the `Keyboard`, the `Video Display Monitor`, a `Character Printer`, and some type of `Pointing Device`.

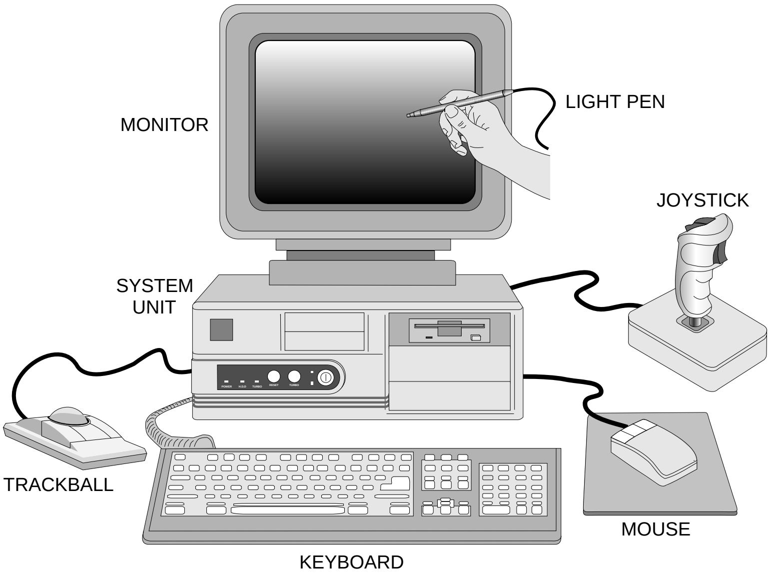
The most widely used input device for personal computers is the typewriter-like alphanumeric keyboard. Unlike other I/O devices, the keyboard normally requires no interface adapter card. Its interface circuitry is generally built directly into the system’s main board.

Pointing devices are small input devices that enable the user to interact with graphical software running in the system. They enable the user to move a cursor, or some other screen image, around the display screen and choose options from an onscreen menu, rather than typing in commands from a keyboard. Most pointing devices are hand-held units that enable the user to enter commands and data into the computer more easily than is possible with a keyboard. Because they make it easier to interact with the computer than other input devices do, they are, therefore, friendlier to the user.

Common input devices are depicted in Figure 1.22.A. The devices include the keyboard, mouse, joystick, light pen, and trackball.

Figure 1.22.A

Typical input devices.



The most widely used pointing device is the `mouse`. Mice are hand-held devices that produce input data by being moved across a surface, such as a desktop. They may have 1, 2, or 3 buttons that can be pressed in different combinations to interact with software running in the system. The mouse enables the user to move a cursor, or some other screen image, around the display screen. When a position has been selected, clicking one or more of the mouse buttons enables the user to choose options from an on-screen menu, rather than type in commands from a keyboard.

Specialized graphics software enable the user to operate the mouse as a drawing instrument. In this context, the mouse can be used to create elaborate pictures on the screen. The `Trackball Mouse` detects positional changes through a rolling trackball on which it rides.

The most widely used output device for personal computers is the `Cathode Ray Tube` (CRT) video display monitor, similar to the one shown in Figure 1.22.B. Video display monitors often include a tilt/swivel base that enables the user to position it at whatever angle is most comfortable. This offers additional relief from eye-strain by preventing the user from viewing the display at an angle. Viewing the screen at an angle causes the eyes to focus separately, which places strain on the eye muscles.

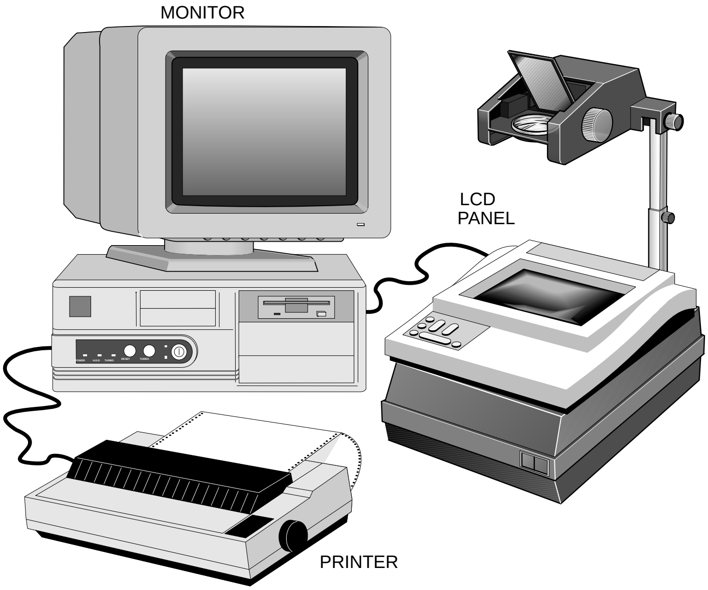


Figure 1.22.B

Typical output devices.

Small notebook and laptop computers use non-CRT displays, such as `Liquid Crystal Display` (LCD) and `gas-plasma panels`. These display systems are well suited to the portability needs of these computers. They are much lighter and more compact than CRT monitors and require much less electrical energy to operate. Both types of display units can be operated from batteries.

After the monitor, the next most frequently added output device is the `character printer`, which produces hard copy output on paper. It converts text and graphical data from the computer into print on a page.

Software

Objectives

After the system's components are connected together and their power connectors have been plugged into a receptacle, the system is ready for operation. However, one thing is still missing: the software. Without good software to oversee its operation, the most sophisticated computer hardware is worthless.

Three general classes of software actually need to be discussed:

- System software
- Applications software
- Games and learning software

The bulk of the software discussed in this book deals with the system software category. This is because this type of software tends to require more technical skills to manipulate and, therefore, most often involves the service person.

System Software

Objectives

The System Software category consists of special programs used by the system itself to control the computer's operation. Two classic examples of this type of software are the system's `Basic Input/Output System` (BIOS) program and the `Disk Operating System` (DOS). These programs control the operation of the other classes of software. The BIOS is located in a ROM IC device on the system board. Therefore, it is commonly referred to as `ROM BIOS`. The DOS software is normally located on a magnetic disk.

Basic Input/Output Systems

When a PC is turned ON, the entire system is reset to a predetermined starting condition. From this state, it begins carrying out software instructions from its BIOS program. This small program is permanently stored in the ROM memory ICs located on the system board. The information stored in these chips is all the inherent intelligence that the system has to begin with.

A system's BIOS program is one of the keys to its `compatibility`. To be IBM PC-compatible, for example, the computer's BIOS must perform the same basic functions that the IBM PC's BIOS does. However, because the IBM BIOS software is copyrighted, the compatible's software must accomplish the same results that the original did, but it must do it in some different way.

During the execution of the BIOS firmware routines, three major sets of operations are performed. First, the BIOS performs a series of diagnostic tests (called POST or `Power-On Self-Tests`) on the system, to verify that it is operating correctly.

If any of the system's components are malfunctioning, the tests cause an error message code to be displayed on the monitor screen, and/or an audio code to be output through the system's speaker.

The BIOS program also places starting values in the system's various programmable devices. These intelligent devices regulate the operation of different portions of the computer's hardware. This process is called `initialization`. As an example, when the system is first started, the BIOS moves the starting address and mode information into the DMA controller. Likewise, the locations of the computer's interrupt handler programs are written into the interrupt controller. This process is repeated for several of the microprocessor's support devices so that they have the information they need to begin operation.

Finally, the BIOS checks the system for a special program that it can use to load other programs into its RAM. This program is called the `Master Boot Record`. The boot record program contains information that enables the system to load a much more powerful control program, called the Disk Operating System, into RAM memory. After the Operating System has been loaded into the computer's memory, the BIOS gives it control over the system. In this way, the total "intelligence" of the system is greatly increased over what was available with just the ROM BIOS program alone. From this point, the Operating System oversees the operation of the system.

This operation is referred to as `booting up` the system. If the computer is started from the OFF condition, the process is referred to as a `cold boot`. If the system is restarted from the ON condition, the process is called a RESET, or a `warm boot.`

The bootup process may take several seconds to perform depending upon the configuration of the system. If a warm boot is being performed, or if the POST tests have been disabled, the amount of time required to get the system into operation is greatly decreased.

In Part 1 of the operation, the BIOS tests the microprocessor and the system's RAM memory . In Part 2, it furnishes starting information to the system's microprocessor support devices , video adapter card , and disk drive adapter card . Finally, in Part 3, the BIOS searches through the system in a predetermined sequence for a master boot record to which it turns over control of the computer. In this case, it checks the floppy disk drive first and the hard disk drive second . If a boot record is found in either location, the BIOS moves it onto the computer's RAM memory and turns control over to it.

Operating Systems

`Operating Systems` are programs designed to control the operation of a computer system. As a group, they are easily some of the most complex programs devised. Every portion of the system must be controlled and coordinated so that the millions of operations that occur every second are carried out correctly and on time. In addition, it is the job of the operating system to make the complexity of the personal computer as invisible as possible to the user. Likewise, the operating system acts as an intermediary between software applications that are nearly as complex, and the hardware on which they run. Finally, the operating system accepts commands from the computer user, and carries them out to perform some desired operation.

A `Disk Operating System (DOS)` is a collection of programs used to control overall computer operation in a disk-based system. These programs work in the background to enable the user of the computer to input characters from the keyboard, to define a file structure for storing records, or to output data to a monitor or printer. The disk operating system is responsible for finding and organizing your data and applications on the disk.

The disk operating system can be divided into three distinct sections:

- Boot files, which take over control of the system from the ROM BIOS during startup
- File management files, which enable the system to manage information within itself
- Utility files, which enable the user to manage system resources, troubleshoot the system and configure the system

The operating system acts as a bridge between the application programs and the computer, as described in Figure 1.23. These application programs enable the user to create files of data pertaining to certain applications such as word processing, remote data communications, business processing, and user programming languages.

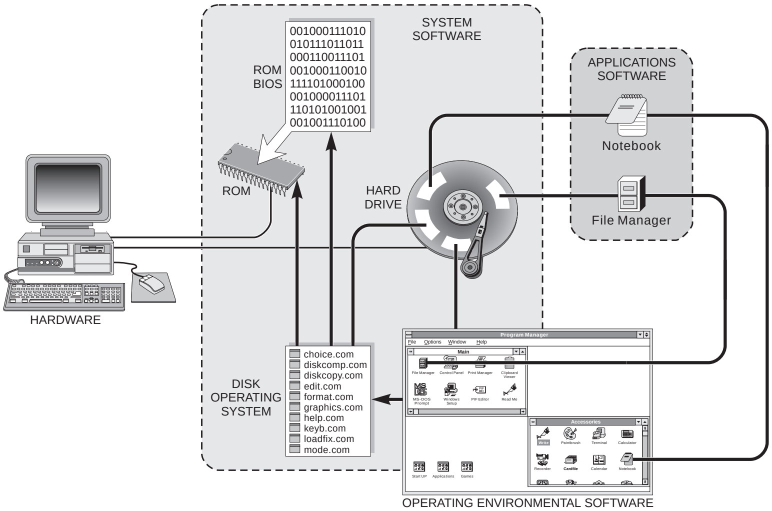


Figure 1.23

The position of DOS in the computer system.

Often, new users are confused because data or programs they have created do not appear from the DOS prompt. The information has been created by an application program that applies its own formatting code to the body of the information. Therefore, the creating program is necessary to properly interpret the created program. As an example, when a text file is created with a given word processing program, formatting codes are added to the document for bold and italic characters, as well as tabs and carriage returns. Although the basic ASCII characters may appear correctly, the codes may be displayed much differently by another word processor. Likewise, there is almost no chance that a BASIC programming package will interpret the control codes so that it can display the text file in any meaningful way.

Operating Environments

Objectives

Another form of operating environment referred to as a `Graphical User Interface` (GUI) has gained widespread popularity in recent years. GUIs, like the one depicted in Figure 1.24, use a graphical display to represent procedures and programs that can be executed by the computer. These programs routinely use small pictures, called `icons,` to represent different programs. The advantage of using a GUI is that the user doesn't have to remember complicated commands to execute a program.

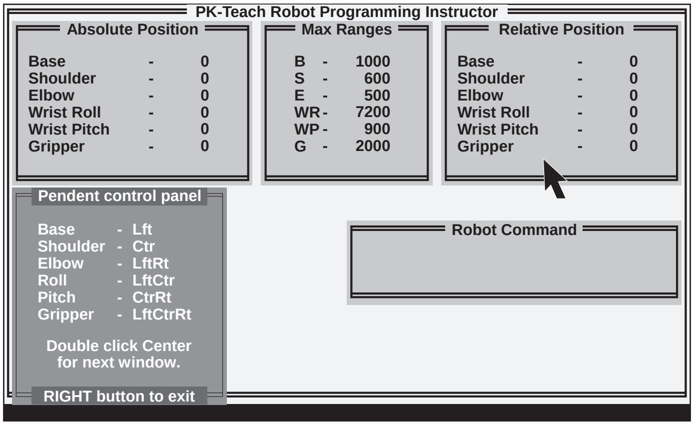


Figure 1.24

A Graphical User Interface screen.

Application Software

Objectives

The second software category, `applications software`, is the set of programs that perform specific tasks, such as word processing, accounting, and so forth. This type of software is available in two forms:

- Commercially available, user-oriented applications packages, which may be bought and used directly
- Programming languages, with which you can write your own applications programs

Commercial Application Packages

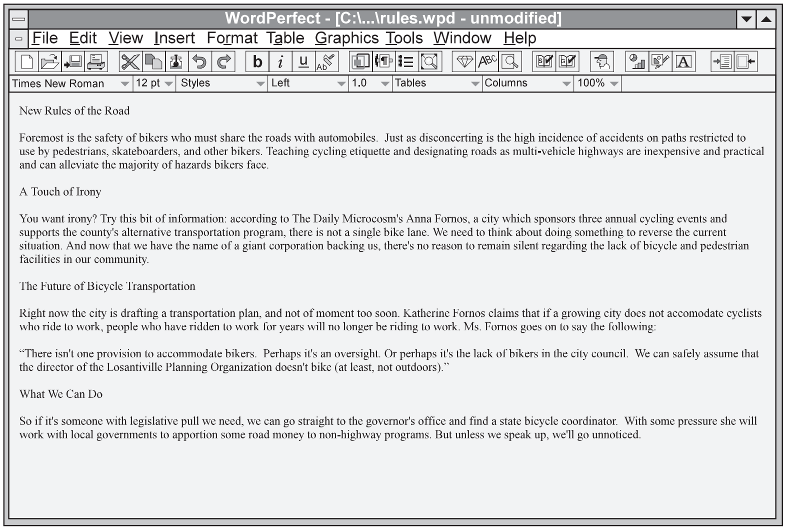
The openness of the personal computer market has generated a wide variety of different applications programs designed for use with them. Even a short discussion of all the software types available for the PC would take up more space than this chapter can afford. However, a small group of these programs make up the vast majority of the software sold in this category. These programs are:

- Word processors
- Spreadsheets
- Database management systems (DBMS)
- Personal productivity tools
- Graphic design packages

Word processors are specialized software packages that can be used to create and edit alphanumeric texts, such as letters, memos, contracts, and other documents. These packages convert the computer into a super typewriter. Unlike typewriters, word processors enable the user to edit, check, any correct any errors electronically before the document is committed to paper. Many word processors offer extended functions such as spelling checkers, as well as on-line dictionary and thesaurus functions that aid the writer in preparing the document. A typical word processor working page is depicted in Figure 1.25.

Figure 1.25

A typical Word Processor.



Spreadsheets are specialized financial worksheets that enable the user to prepare and manipulate numerical information in a comparative format. Paper spreadsheets were used by business people for many years before the personal computer came along. Spreadsheets are used to track business information such as budgets, cash flow, and earnings. Because the information on these types of documents is updated and corrected often, working on paper was always a problem. With electronic spreadsheets, such as the one illustrated in Figure 1.26, this work is much quicker to perform and less fatiguing. This type of software may be the class most responsible for the growth of personal computers into serious work machines.

Figure 1.26

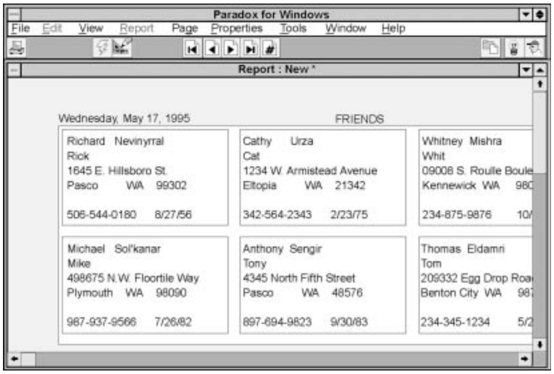
A typical spreadsheet program.

	A	B	C	D	E
3	**ISBN: 1-56205-896-7**				**Page Count = 750**
4					
5	**Element**	**Title**	**Author**	**IN**	**# Pgs**
6	IFC	Inside Front Cover/Tear Card			
7					
8	1	Microcomputer Fundamentals	CB	X	59
9	2	PC Hardware	CB	X	43
10	3	Operating Systems	CB	X	57
11	4	Microsoft Windows	CB	X	46
12	5	Troubleshooting the System	CB	X	60
13	6	System Boards	CB	X	70
14	7	Input/Output	CB	X	63
15	8	Magnetic Storage	CB	X	47
16	9	Video Displays	CB	X	52
17	10	Printers	CB	X	50
18	11	Data Communications	CB	X	30
19	12	Multimedia	CB	X	32
20	13	Preventative Maintenance and Safety	CB	X	35
21	14	Customer Satisfaction	CB	X	15
22		Appendices			
23	A	Overview of the A+ Certification			
24	B	Study Tips			
25	C	What's on the CD-ROM			
26	D	Glossary			
27					

Database management systems (or simply Databases) are programs that enable the user to store and track vast amounts of related information about different subjects. Databases can be thought of as electronic boxes of note cards. On these electronic note cards you can keep several pieces of information related to a given subject. For example, you might keep certain types of information on a note card for each of your relatives. The card might contain such information as a phone number, address, and birthday. The database enables you to sort through the information in different ways. With a database program it would be no problem to identify all the relatives that have a birthday in a given month. A typical database working page is shown in Figure 1.27.

Figure 1.27

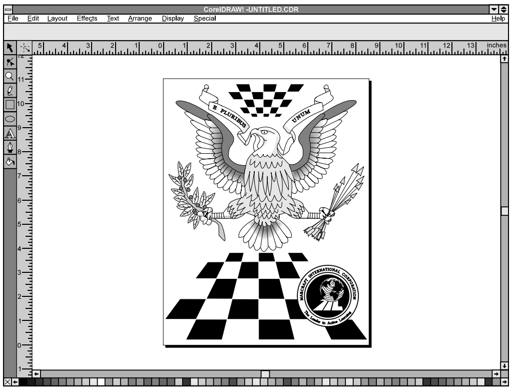
A typical database program.



Graphics programs enable the user to create non-alphanumeric output from the computer. Simple graphics programs are used to create charts and graphs that represent data. More complex programs can be used to create artistic output in the form of lines, shapes, and images. Typically, graphic design programs produce graphics in two types of formats: as `bit-mapped images` and as `vector images`. With bit-mapped graphics, every dot (pixel) in the image is defined in memory. Vector images, on the other hand, are defined as a starting point and a set of mathematical formulas in memory. Because vector images exist only as a set of mathematical models, their size can be scaled up or down easily without major distortions. Vector images can also be rotated easily, enabling three-dimensional work to take place on these images. On the other hand, bit-mapped graphics are tightly-specified collections of spots across and down the screen. These types of images are very difficult to scale or rotate without distortion. A typical graphics package is depicted in Figure 1.28.

Figure 1.28

A typical graphic design program.



Personal productivity programs, also referred to as `desktop organizers`, encompass a variety of programs that simulate tools found on typical business desks. They normally include items such as telephone directories, calculators, note pads, and calendar programs. Of course, many other types of applications software are available for use with the PC. Any type of meaningful discussion of all these software types is well beyond the scope of this book, and certainly goes well beyond the scope of preparing for A+ testing.

Programming Packages

Because the only language that computers understand is their own machine language, and most humans don't relate well to machine languages, you need a piece of system software to convert whatever language you're using into the machine's language. These conversion packages exist in two forms: `Interpreters` and `Compilers`. The distinction between the two is in how and when they convert the user language into machine language. Interpreters convert the program as it is being run (executed). Compilers convert the entire user-language program into machine code before it is executed. Typically, compiled language programs execute much faster than those written in interpretive languages. In addition, compiled languages typically provide the user with a much higher level of direct control over the computer's operation.

In contrast, interpreted languages are usually slower, and less powerful, but their programs tend to be easier to write and use than compiled languages.

BASIC is probably the best-known example of an interpreted `high-level` programming language. The term BASIC stands for `beginners all-purpose symbolic instruction code`, while "high-level" refers to the fact that the language uses commands that are English-like. This all contributes to making BASIC a popular user language, which almost anyone can learn to use.

An example of a compiled language is FORTRAN. FORTRAN is one of the oldest user languages still in use today, and is primarily used in engineering and scientific applications. Many of FORTRAN's attributes resemble those you find in BASIC. This is because most versions of BASIC are modified derivatives of FORTRAN.

BASIC compiler programs are also commercially available that enable you to take BASIC programs, written with an interpreter, and compile them so that they run faster. Other compiled language packages that run on almost any PC include: COBOL (a business applications language), LISP (an artificial intelligence applications language), and the popular user languages C+, FORTH and PASCAL.

Another alternative in programming your computer is to write programs in `Assembly language` (one step away from machine language), and run them through an `Assembler` program. Assembly language is a human-readable form of machine language that uses short, symbolic instruction words, called `mnemonics`, to tell the computer what to do. Each line of an Assembly language program corresponds directly to one line of machine code. Writing programs in Assembly language enables the programmer to precisely control every aspect of the computer's operation during the execution of the program. This makes Assembly language the most powerful programming language you can use. To its detriment, Assembly language is complex, and requires the programmer to be extremely familiar with the internal operation of the system using the program.

A number of steps are required to create an Assembly language program:

1. First, you must create the program using an alphanumeric text editor.
2. Next, the text file must be run through an Assembler program, to convert the Assembly language into machine code.
3. Finally, the machine code must be run through a `linking program`, which puts the assembled machine code into the proper format to work with the operating system. In this final form, the program can be executed from DOS.

For short and simple Assembly language programs, a DOS utility called `DEBUG` can be used to enter and run machine language and limited Assembly language programs, without going through the various assembly steps.

Microsoft introduced a radically different programming environment when it delivered `Visual Basic`. Unlike the earlier BASIC language versions, Visual Basic is a graphical programming tool that enables programmers to develop Windows applications in an artistic rather than command-line basis. The programmer draws graphic elements and places them on the screen as desired. This tool is so powerful that it is used to produce large blocks of major applications, as well as finished Windows products. The finished product can be converted into an executable file using a Visual Basic utility. The only major drawback of Visual Basic is that major applications written in it tend to run slowly, because it is an interpreted language.

Games and Educational Packages

Games and learning programs are among the leading titles in retail software sales. The games market has exploded as PC speeds have increased, and as output graphics have improved. However, on the technical side, there is generally not much call for a repair associate with game software. Most games work with well-developed pointing devices, such as trackballs and joysticks, as the primary input devices. Although the housing designs of these products can be quite amazing, they tend to be simple and well-proven devices, requiring relatively little maintenance. Likewise, the software tends to be pretty straightforward from a user's point of view. It simply gets installed and runs.

`Computer-aided instruction` (CAI) and `computer-based instruction` (CBI) have become accepted means of delivering instructional materials. In CAI operations, the computer assists a human instructor in delivering information and tracking student responses. In CBI operations, the computer becomes the primary delivery vehicle for instructional materials.

As these teaching systems proliferate, more complex input, output and processing devices are added to the system. A basic teaching system requires a minimum of a sound card, a fast hard drive, a CD-ROM drive, and a high-resolution video card. Beyond this, CAI and CBI systems may employ such wide-ranging peripherals as large LCD display panels, VGA-compatible overhead projectors, intelligent white boards, wireless mice, and touch-sensitive screens as input devices, full-motion video capture cards, and a host of other multimedia related equipment. Refer to Chapter 12, “Multimedia,” for more information on these types of products.

Version Numbers

All types of software are referred to by version numbers. When a programmer releases a software program for sale, a version number is assigned to it, such as Windows 3.11 or MS-DOS 6.22. The version number distinguishes the new release from any prior releases of that same software. The larger the version number, the more recent the program is. When new features or capabilities are added to a program, it is given a new version number. Therefore, referring to a software package by its version number indicates its capabilities and operation. The number to the left of the decimal point is the major revision number, which usually changes when new features are added. The number(s) to the right of the decimal are minor revision numbers, which usually change when corrections are made to the program.

Summary

This chapter has presented a mini-course on the basic organization and operation of the digital computer. The initial section of the chapter described the various types of information and words used in computer systems. The following section examined the basic hardware building blocks that make up the typical microcomputer. The final section of the chapter looked at different classes of software. Types of system software were examined first. This was followed by a discussion of typical commercial software packages that enable users to adapt the system for work on a wide variety of applications.

After you have completed this chapter, you should be prepared to investigate the components presented in this chapter in much greater detail. As you move through the remainder of the text, you may refer back to the materials developed in this chapter and, in many cases, refine the concepts initiated here.

The next chapter builds on the fundamental materials from this chapter by demonstrating how these conceptual topics are implemented in actual personal computer equipment.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand and can perform each item listed there. Afterward, answer the Review questions that follow to verify your knowledge of the information.

Lab Exercises

Hands-on lab procedures correspond to the theory materials presented in this chapter. Refer to the lab manual and perform Procedures 14, “QBASIC,” and 16 “Advanced QBASIC.”

Also, perform the following `paper labs` to become familiar with the basic operation of the computer system:

1. Use the $1.98 Computer’s instruction set to write a program for it that loads a value of 7 from location F into the accumulator. It should then add 2 to the value in the accumulator, compare the result to a value of F, and set the E flag if the two values are equal. Continue to add 2 to the value in the accumulator until the E flag is set. At this time, the program should halt.

2. Using the $1.98 Computer’s instruction set, write a program for it that multiplies 7 by 6 and stores the final answer in location F. (Consider the accumulator to be 8 bits wide.)

3. Using the $1.98 Computer’s instruction set, write a program for it that divides the decimal value 105 by 15 and places the answer in location F.

4. Using the $1.98 Computer’s instruction set, write a program for it that counts by 3’s to a value of F and then begins the count again. After counting to F five times, cause the program to Halt.

Lab Answers

1. AF, B8, 6C, 7E, 21, 00, 00, 00, 02, 00, 00, 00, 0F, 00, 50, 07

2. BB, 4F, AC, CD, 4C, 6A, 7E, AF, 20, 00, 00, 07, 06, 01, 50, 00

3. AB, CC, BF, AD, BF, 4F, AB, 6A, 7E, 21, 00, 69, 0F, 01, 50, 00

4. AB, BB, 6C, 75, 21, AA, CD, 3E, 20, 00, 05, 03, 0F, 01, 50, 00

Review Questions

The following questions test your knowledge of the material presented in this chapter.

1. What does BCD stand for?
2. Describe the differences between an EPROM and EEPROM.
3. Which control bus signal is used to synchronize all microprocessor operations?
4. What is another name for an 8-bit word?
5. What determines where a data word is stored in a memory unit?
6. What type of operation is being performed when a data word is placed into a memory unit?
7. Define volatile memory.
8. What is the major disadvantage of dynamic RAM? Of primary memory?
9. What is the major advantage of dynamic RAM?
10. What is the major requirement for devices used as internal, or primary memory?
11. What are the major requirements for secondary memory devices or systems?
12. Name the basic blocks of a typical computer.
13. What components are incorporated to form a microprocessor?
14. Where do the contents of the microprocessor's internal registers go during an interrupt operation?
15. What are the three major differences between floppy-disk drives and hard-disk drives?

16. Which bus in the computer is bi-directional and which is uni-directional?

17. What type of memory device is used to hold data that does not change?

18. Which software category is normally associated with the term "firmware"?

19. Define "files" as they apply to computers.

20. What function do the expansion slots perform in a typical microcomputer?

21. What three functions do the ROM BIOS programs perform?

22. What is the purpose of a GUI?

23. Why is magnetic disk data storage more popular than tape storage?

24. How is magnetic tape storage different than magnetic disk storage?

25. Describe two common formats used to store data on magnetic tape.

Review Answers

1. Binary-Coded Decimal. For more information, see the section "Numeric Data Words."

2. An EPROM can be erased and re-written using an external erasing/writing system, while EEPROMs can be reprogrammed by special circuitry included in the computer system. For more information, see the section "ROM Memory."

3. The system clock signal. For more information, see the section "Control Bus."

4. Byte. For more information, see the section "Bits, Bytes, and Computer Words."

5. Its address. For more information, see the section "Memory Unit."

6. A Write or Store operation. For more information, see the section "Memory Unit."

7. Memory that requires that power be applied to it in order to retain data. For more information, see the section "RAM Memory."

8. It requires extra circuitry to keep it refreshed. This adds to the initial cost of the memory unit. For more information, see the section "RAM Memory."

9. Its high circuit density (or storage capacity) and its low cost of operation. For more information, see the section "RAM Memory."

10. Speed. It must be fast enough to work without slowing down the microprocessor. For more information, see the section "Memory Unit."

11. Low cost and non-volatility. For more information, see the section "Memory Unit."

12. Input, Output, Memory, ALU, and Control units. For more information, see the section "Basic Computer Structure."

13. ALU and Control units along with internal registers. For more information, see the section "Microprocessors Defined."

14. To a special section of memory called the `stack`. This can be an area of primary RAM memory, called a software stack, or a specialized set of internal microprocessor registers, called a hardware stack. For more information, see the section "Internal Registers."

15. Storage Capacity, data transfer rates, cost. For more information, see the section "Hard-Disk Drives."

16. The data bus is bi-directional, and the address bus is uni-directional. For more information, see the sections "Data Bus" and "Address Bus."

17. ROM. For more information, see the section "Memory Unit."

18. System Software. For more information, see the sections "ROM Applications" and "System Software."

19. Logical groups of information that are stored and treated as a single unit. For more information, see the section "Floppy-Disk Drives."

20. The expansion slots provide uniform connection points that enable I/O interface cards to be plugged into the system's address, data, and control buses. For more information, see the section "Expansion Slot Connectors."

21. Power-on self tests, initialization of intelligent devices, and disk drive boot up. For more information, see the section "Basic Input/Output Systems."

22. To enable the user to choose and execute functions from a graphical display rather than use command line entries. For more information, see the section "Operating Environments."

23. Access to data stored on a magnetic disk is direct, while access to data stored on tape is sequential. Therefore, disk access to data is much faster on disk than it is on tape. For more information, see the section "Secondary Memory."

24. Data is stored on tape in a sequential manner along the length of the tape. To access the 33rd record on a tape, the preceding 32 records must pass under the R/W heads before it can be read or written. In most other respects, data storage on tape is very similar to that on disk. For more information, see the section "Secondary Memory."

25. Tape storage is generally performed in multiple parallel tracks along the tape, or in a serpentine fashion that records on multiple tracks but in opposite directions along the tape. Parallel track tape machines require a R/W head for each track while a serpentine recorder requires only one R/W head. For more information, see the section "Secondary Memory."

Chapter 2

PC Hardware

Upon completion of this chapter and its related Lab Exercises, you should be able to:

Objectives

- Discuss differences between different PC case styles and explain strong and weak points associated with each.
- Recognize the special features associated with portable computers.
- Describe the function of typical PC power supplies.
- Identify different types of RAM modules (DIP, SIPP, SIMM, DIMM).
- Identify common microprocessor IC package types.
- Discuss and recognize the different PCMCIA devices currently available.
- Identify a Video/Graphics/Array (VGA) adapter card.
- Identify a Multi I/O (MI/O) card.
- Recognize different disk drive types associated with PCs.
- Describe typical external connections associated with PCs.
- Properly connect the keyboard, monitor, and AC power cord to the system.
- Locate the power supply unit, system board, system speaker, disk drive unit, and expansion slots.

continues

- Locate the system's RAM banks and, using documentation, determine the amount of RAM installed in the system.
- Install basic system hardware and peripheral devices.
- Configure basic system hardware.
- Disassemble and reassemble a system.

Introduction

In the preceding chapter, basic digital computer concepts were introduced. This chapter illustrates how those basic concepts have been used to develop actual microcomputer systems. The initial portion of the chapter documents the evolution of the personal computer from a variety of small computing systems that started the market, to the Apple and IBM personal computers that brought PCs to prominence, and finally to the clones that currently dominate the market.

The remainder of the chapter deals with the various hardware structures that make up a PC-compatible microcomputer. This section lays the foundation for the more in-depth theory and troubleshooting materials in the following chapters.

Personal Computer Evolution

In the early years of microcomputer history, the market was dominated by a group of small companies that produced computers intended mainly for playing video games. Serious computer applications were almost always a secondary concern. Companies such as Commodore, Timex/Sinclair, Atari, and Tandy produced 8-bit machines based on microprocessors from Intel, Motorola, Zilog, and Commodore for the emerging industry.

In 1977, Apple Computers produced the Apple I. This was followed with a series of 8-bit microcomputers: the Apple II, Apple IIc, and Apple IIe. These units were single-board computers with built-in keyboards and a discrete monitor. With the IIe unit, Apple installed seven expansion connectors on its main board. These connectors were included to enable adapter cards to be added to the system. Apple also produced a set of adapter cards that could be used with the IIe to bring additional capabilities to the basic IIe. These units were very advanced for their time. However, in 1981, Apple introduced their very powerful 16-bit Macintosh (Mac) system to the market. The Mac features represented a major shift in computing power. They departed from command-line operations by offering a graphical operating method that used a small input device called a mouse to select objects from the output monitor and guide the operation of the system. This method found an eager audience of nontechnical users and public school systems.

Late in 1981, IBM entered the personal computer market with the unveiling of their now famous IBM-PC. At the time of its introduction, the IBM-PC was a drastic departure from the status quo of the microcomputer world. The PC used an Intel 8088 16/8-bit microprocessor (its registers process data 16 bits at a time internally, but have an 8-bit external data bus). Relatively speaking, it was fast, powerful, flexible, and priced within the range of most individuals. The general public soon became aware of the tremendous possibilities of the personal computer, and the microcomputer world quickly advanced from one of simple games to one with a seemingly endless range of advanced personal and business applications.

The original Apple Mac and IBM PC are depicted in Figure 2.1.

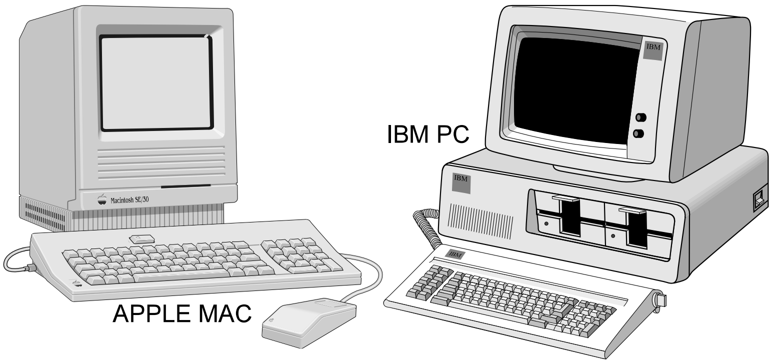


Figure 2.1

The Apple Mac and IBM PC.

The success of the PC stemmed largely from IBM's approach to marketing it. IBM actually constructed very little of the PC. Instead, they went to independent manufacturers for most of their system components and software. To do this, IBM made public nearly all the technical information concerning the PC. This openness, in turn, created a rush of small companies developing hardware and software options compatible with the PC.

In 1983, IBM added a small hard-disk drive to the PC and introduced the Extended Technology (XT) version of the PC. The success continued when, in 1984, IBM introduced the Advanced Technology PC (PC-AT). The AT used a true 16-bit microprocessor (it processes 16 bits at a time internally and has a 16-bit external data bus) from Intel, called the 80286. The wider bus increased the speed of the computer's operation, because the 80286 was then able to transfer and process twice as much data at a time as the 8088 could.

The IBM PC-AT is depicted in Figure 2.2. The tremendous popularity of the original IBM PC-XT and AT systems created a set of Pseudo Standards for hardware and software compatibility. The AT architecture became so popular that it has become the industry standard architecture (ISA). The majority of microcomputers are both hardware- and software-compatible with the original AT design.

Figure 2.2

The IBM PC-AT.

Since the days of the original AT design, Intel has introduced several different microprocessors, such as the 80386, the 80486, the Pentium (80586), the Pentium Pro (80686), and Pentium II.

In response to microprocessor clone manufacturers using the 80x86 nomenclature, Intel dropped the numbering system and adopted the Pentium name so that they could copyright it. More inclusive information about microprocessor numbering is presented in the Microprocessor section of Chapter 6, "System Boards."

All of these microprocessors are upwardly compatible with the 8088 design. This means that programs written specifically for the 8088 can be executed by any of the other microprocessors. However, a program written specifically for an 80386 cannot be executed by using an 8088 or 80286. The majority of microcomputers today are based on the AT design but incorporate the newer microprocessors into the system.

With so many PC-compatible software and hardware options on the market, a number of independent companies have developed PC-like computers of their own. These are referred to as `PC look-alikes`, `clones`, or more commonly, `compatibles`. The cloning process was made possible by two events. The government-backed `Electronic Research and Service Organization` (ERSO) in Taiwan successfully produced a noncopyright infringing version of the XT BIOS firmware and IBM did not lock up exclusive rights to the Microsoft Disk Operating Software that controls the interaction between the system's hardware and the software applications running on it.

In an attempt to derail the growing market acceptance of clone PCs, IBM introduced a new line of `Personal System/2` (PS/2) computers, which included a new, patented 32-bit expansion bus standard called `Micro Channel Architecture` (MCA). The Personal System/2 (PS/2) Desktop is depicted in Figure 2.3.

The Micro Channel Architecture, which was invented, trademarked, and patented by IBM in 1987, provided computing potential that was more closely related to larger mainframe computers than contemporary microcomputers. The advanced design of MCA made provisions that enabled systems using its architecture to work on several tasks at one time and to include several microprocessors working together as a unit.

Figure 2.3

Personal System/2 desktop unit.

In developing MCA, IBM disregarded any efforts to remain compatible with the ISA standard architecture. Indeed, the MCA standard stands alone in terms of hardware compatibility. The physical expansion connector is incompatible with the edge connectors on ISA adapter cards. In addition, the interface signals called for in the MCA standard are different in definition as well as layout. Therefore, the only source for PS/2 peripheral equipment is IBM or one of its approved vendors. Conversely, IBM has maintained software compatibility between its PS/2 line and the earlier PC, XT, and AT lines.

The majority of the discussion in this text deals with PC-compatible designs because they are the most wide-ranging and because they occupy such a large portion of the personal computer market. You may also notice that most of the discussions tend to lean toward desktop PCs. This information should not be difficult to transfer to other PC styles.

The PC System

Objective

A typical personal computer system is modular by design. It is called a system because it includes all the components required to have a functional computer:

- Input devices—keyboard and mouse
- Computer—system unit
- Output devices—a CRT monitor and a character printer

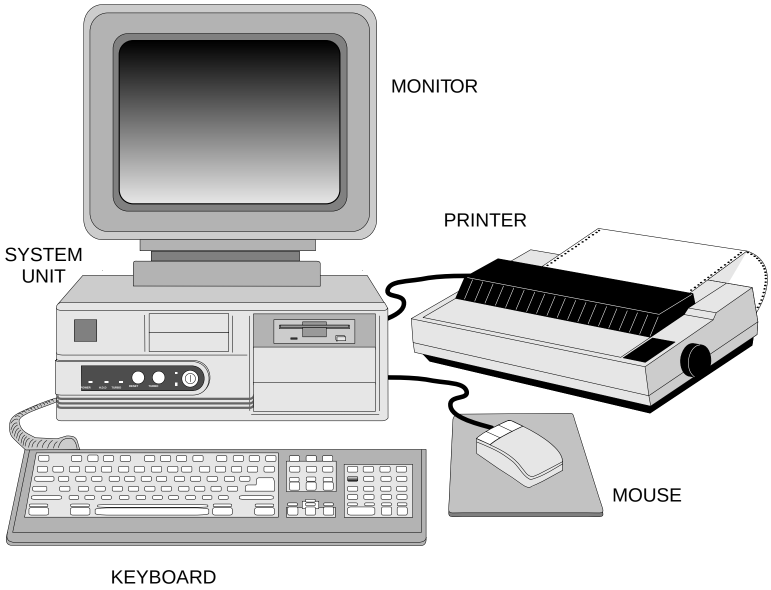


Figure 2.4

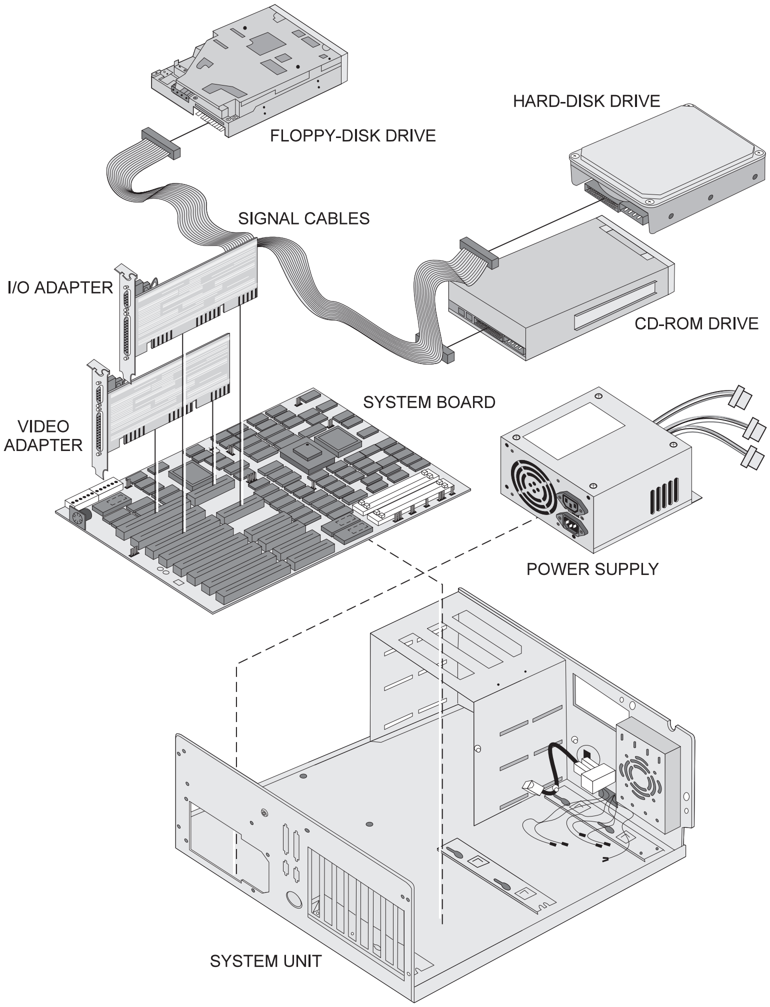
A typical PC system.

The `system unit` is the main portion of the microcomputer system and is the basis of any PC system arrangement. The components surrounding it vary from system to system depending upon what particular functions the system is supposed to serve.

The components inside the system unit can be divided into four distinct sub-units: a switching Power Supply, the Disk Drives, the System Board, and the Options Adapter cards, as illustrated in Figure 2.5.

Figure 2.5

The components in a system unit.



A typical system unit contains a single `power supply unit` that converts commercial power into the various levels required by the different units in the system. The number and types of disk drives in the system vary according to the application for which the system is designed. However, a single `floppy-disk drive unit`, a single `hard-disk drive unit`, and a single `CD-ROM drive` are typically installed to handle the system's mass storage requirements.

The `system board` is the center of the system. It contains the portions of the system that define its computing power and speed. System boards are also referred to as `motherboards`, or `planar boards`. Any number of options adapter cards may be installed to handle a wide array of PC peripheral equipment. Typical adapter cards installed in a system include a `video adapter card` and some sort of `Input/Output (I/O) adapter card`. Peripheral devices such as printers and mice normally connect to options adapter cards through the rear of the system unit. These cards plug into `Expansion Slot Connectors` on the back of the system board. In most desktop cases, the keyboard also plugs into the back panel.

Cases

Objective

PCs have been built in a number of different case designs. Each design offers characteristics that adapt the system for different environments. These key characteristics for case design include mounting methods for the printed circuit boards, ventilation characteristics, total drive capacity, footprint (the amount of horizontal space they occupy), and portability.

Desktops

Some of the most familiar PC case styles are the desktop designs illustrated in Figure 2.6. These units are designed to sit horizontally on a desk top (hence the name). The original IBM PC, PC-XT, and PC-AT designs use this case style. The PC and XT case styles measured 21"w×17"d×5-1/2"h, while the AT case grew to 23"w×17"d×5-1/2"h.

A narrower desktop style, referred to as the `baby AT case`, was developed to take up less desk space than the XT and AT styles. The reduced footprint was accomplished by using a half-height power supply unit and limiting the number of disk drives. The disk-drive cage did not reach down to the floor of the system unit, enabling the system board to slide under the power supply and disk-drive cage. The widths of baby AT cases varied from manufacturer to manufacturer.

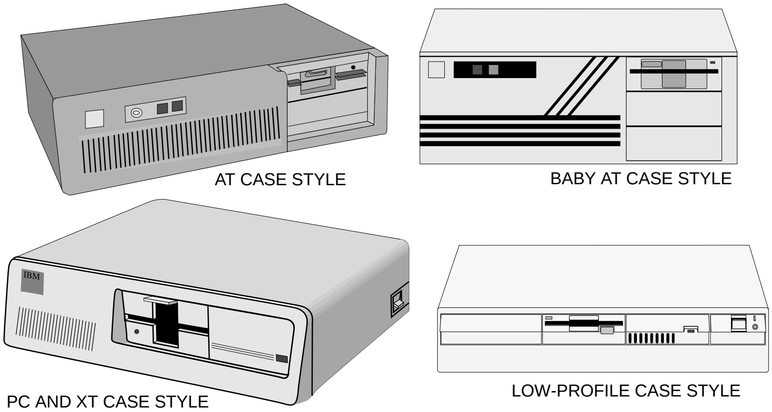


Figure 2.6

Desktop case designs.

A special variety of desktop cases, referred to as `Low-profile Desktops,` reduce the vertical height of the unit. A short bus-extender card, called a `back plane`, mounts in an expansion slot and enables option adapter cards to be mounted in the unit horizontally. This enables the case to be shorter. The horizontal mounting of the I/O cards in a low-profile case tends to create heat build-up problems. The heat rising off of the system board flows around the I/O cards, adding to the heat they are already generating. A standard back plane card is depicted in Figure 2.7.

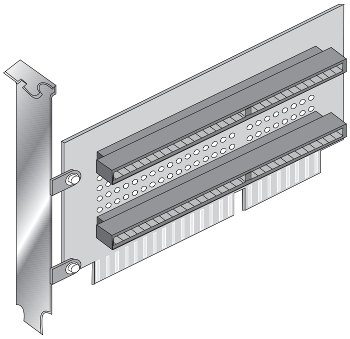


Figure 2.7

A back plane card.

Low-profile power supplies and disk drives are also required to achieve the reduced height. Only the very earliest low-profile units tried to incorporate a 5-1/4" floppy drive into the design. The vast majority of these units have a single 3-1/2" floppy drive only. IBM's original PS/2 units incorporated these smaller components and shorter microchannel adapter cards to achieve a relatively low profile without mounting the cards horizontally.

In desktop cases, the system board is generally located in the floor of the unit, toward the left-rear corner. The power supply is located in the right-rear corner. Raised reinforcement rails in the floor of the system unit contain threaded holes and slip-in slots to which the system board is anchored. Small plastic feet are inserted into the system board and set down in the slots. The system board is secured when its feet are slid into the narrow portion of the slot.

One or two brass standoffs are inserted into the threaded holes before the system board is installed. After the system board has been anchored in place, a small machine screw is inserted through the system board opening and into the brass standoff. This arrangement provides electrical grounding between the system board and the case and helps to reduce `electromagnetic field interference` (EFI) emitted from the board.

Every electrical conductor radiates a field of electromagnetic energy when an electrical current is passed through it. Computer systems are made up of hundreds of conductors. The intensity of these fields increases as the current is turned on and off. Modern computers turn millions of digital switches in their ICs on and off each second, resulting in a substantial amount of energy radiating from the computer. The levels generated by the components of a typical system can easily surpass maximum-allowable radiation levels set by the Federal Communications Commission. The fields generated can potentially interfere with reception of radio, television, and other communications signals under the FCC's jurisdiction. Therefore, computer manufacturers design grounding systems and case structures to limit the amount of EFI that can escape from the case.

Basically, the FCC has established two certification levels for microcomputer systems:

- Class A is a level set for computers in business environments.
- Class B is a stricter level set for sales directed at general consumers for the home environment.

FCC compliance stickers are required on computer units, as is certain information in their documentation.

Looking inside a desktop system unit, as depicted in Figure 2.8, reveals the arrangement of its major components.

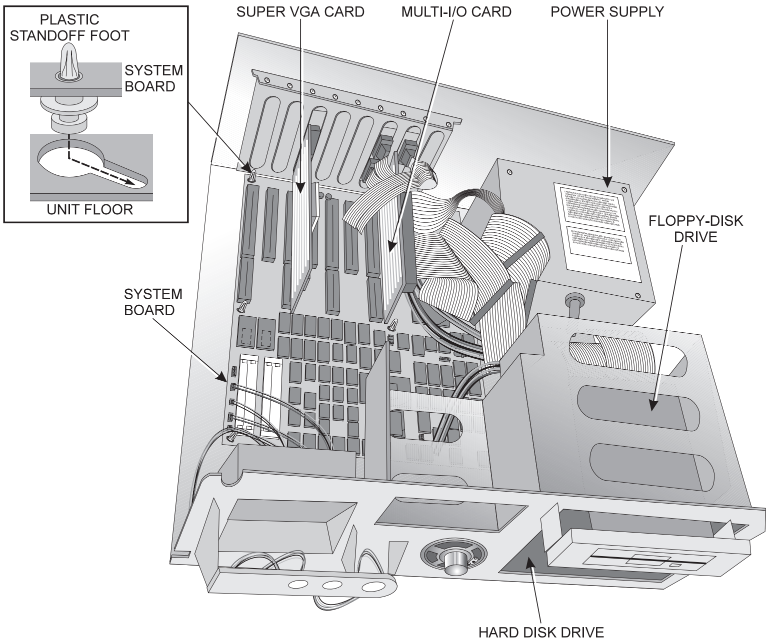


Figure 2.8
Inside a desktop unit.

The disk drive units are located in bays at the right front corner of the system. XT-style units use two side-by-side bays, each capable of handling a `full-height` (3.38"×5.87"×8") 5-¼" drive unit. Newer desktops are designed to hold between 3 and 5 half-height 5-¼" drives. Normally, a fixed drive bay capable of holding 2 or 3 half-height drives is built into the unit. Additional 5-¼" or 3-½" removable drive bays may also be included. The removable bays are secured to the system unit with machine screws and, when removed, provide easier access to the system board. Some cases require that the drive be attached using machine screws; AT-type cases have grooves in the bay that use slide-in mounting rails that are attached to the drive. The drives are held in place by the front panel of the slide-on case. The smaller 3-½" drives require special mounting brackets that adapt them to the 5-¼" bays found in many desktop cases.

System indicator lights and control buttons are built into the front panel. Typical indicator lights include a power light, a hard drive activity light, and a turbo speed indicator light. Control buttons include a power switch, a turbo speed selection switch, and a reset button. Older system units used an ON/OFF flip switch that extended from the power supply at the right rear edge of the case. Newer units place the on/off switch on the machine's front panel and use an internal power cable between the switch and the power supply unit. The power supply's external connections are made in the rear of the unit. The system's installed options adapter cards are also accessed through the system's back panel. Figure 2.9 shows typical front panel controls and indicators.

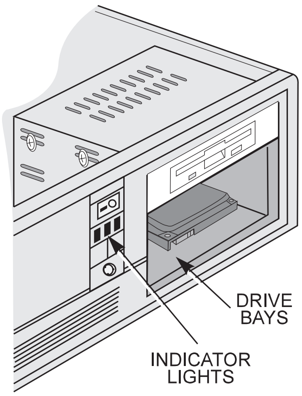


Figure 2.9

Typical front panel controls and indicators.

The upper portion of the system unit slides (or lifts) off the base, as described in Figure 2.10. The tops of some designs slide forward after screws securing it to the back panel have been removed. In this style of case, the plastic front panel usually slides off with the metal top. In other designs, the top swings up from the rear and slides backward to clear the case. The tops of these units are secured to the base by screws in the rear of the unit and screws or clips along the sides of the case. The plastic front panel is attached directly to the case.

The fit between the top and the case is very important in achieving FCC certification. A tight fit and electrical conductivity between the case and top are necessary to prevent unwanted radio interference from escaping the interior of the case.

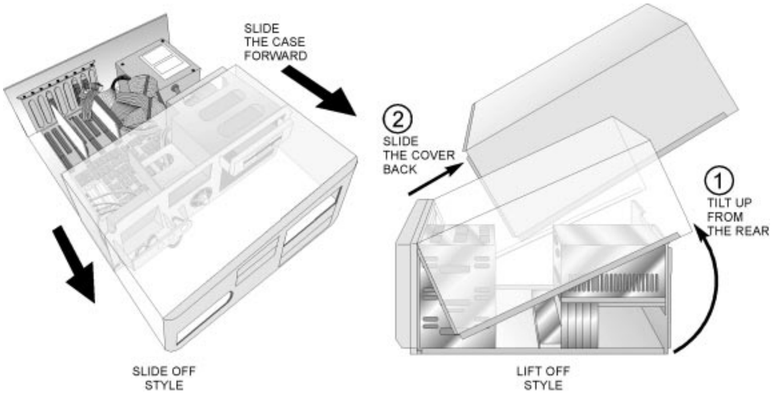


Figure 2.10

Removing cases from desktop units.

The inside face or the plastic front panel is coated with a conductive paint to limit the radio magnetic interference escaping from the case.

A fan in the power supply unit pulls in air through slots in the front of the case. The air flows over the system and options boards, into the power supply unit, and is exhausted through the back of the case. Heat buildup inside the system unit increases as more internal options are added to the system. To compensate for additional heat, it may be necessary to add additional fans to the case. Special `IC cooler fans` are often added to advanced microprocessors. These fans are designed to be fitted directly onto the IC and plug into one of the power supply's connectors.

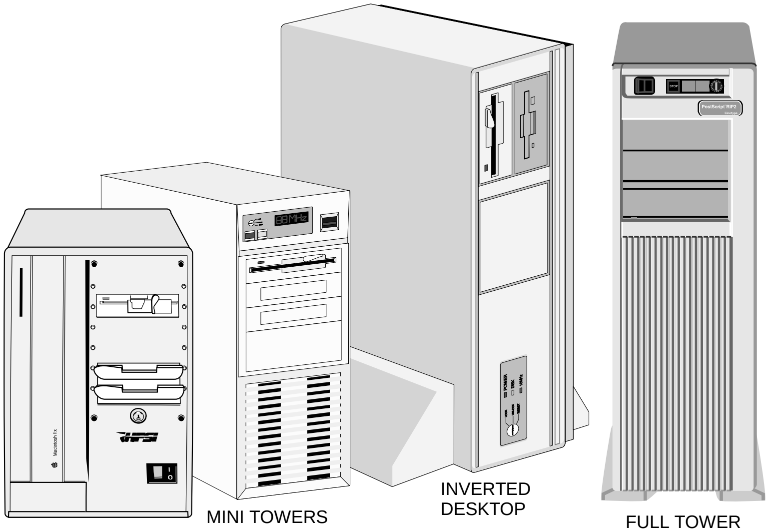
Towers

`Tower cases` are designed to sit vertically on the floor beneath a desk. Some AT users resorted to standing the computers on their sides under the desk to provide more usable workspace on the desktop. This prompted computer makers to develop cases that would naturally fit under the desk. IBM validated the tower design when they introduced the PS/2 models 60 and 80. Different tower case styles are depicted in Figure 2.11.

The system board is mounted to the right side-panel of the case. The power supply unit is attached to the back panel. Indicator lights and control buttons are located toward the upper part of the front panel. The drive units are mounted in the disk drive bays located in the upper half of the front panel.

Figure 2.11

Tower case designs.



Although hard and floppy drives can be mounted on their sides with no real problem—as they were in the adapted AT cases—older drives can lose tracking accuracy when mounted this way. Tower cases enable the disk drives to be mounted in a horizontal fashion. They also offer extended drive bay capabilities that make them especially useful in file server applications where many disk, CD-ROM, and tape drives may be desired.

Many easy-access schemes have been developed to give quick or convenient access to the inside of the system unit. Some towers use removable trays into which the system board and I/O cards are plugged in before being slid into the unit. This enables all the boards to be assembled outside the system unit. Other tower cases use hinged doors on the side of the case that enable the system and I/O boards to swing away from the chassis for easy access.

The ventilation characteristics of most tower units tend to be poor. The reason is associated with the fact that the I/O cards are mounted horizontally. This enables the heat produced by lower boards to rise past the upper boards, adding to the cooling problem. To compensate for this deficiency, most tower units include a secondary case fan to help increase the air flow and dissipate the heat.

`Mini towers` are short towers designed to take up less vertical space. Internally, their design resembles a vertical desktop unit. They are considerably less expensive than the larger towers due to the reduced quantity of materials needed to produce them. Unlike their taller relatives, mini towers do not provide abundant space for internal add-ons or disk drives. However, they do possess the shortcomings of the full towers. Mini towers exist more as a function of marketing than as an application solution.

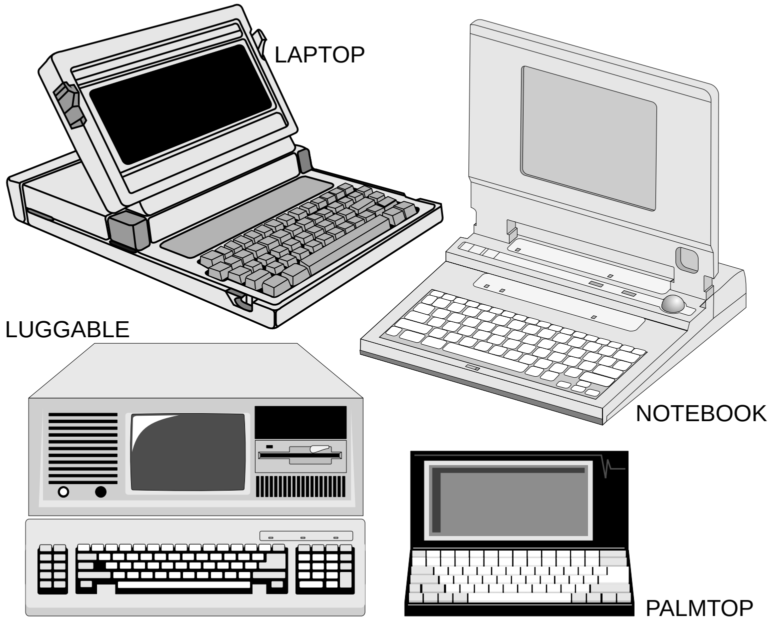
Portables

Objective

To free the computer user from the desk, an array of `portable PCs` have been developed. The original portables were called `luggables`. Although they were smaller than desktop computers they were not truly convenient to transport. The first portables included small, built-in CRT displays and detachable keyboards. The battery and CRT equipment made them extremely heavy to carry. Therefore, they never really had a major impact on the PC market. However, they set the stage for the development of future portable computer systems. Examples of different portable computer designs are shown in Figure 2.12.

Figure 2.12

Some portable computers.



With advancements in battery design and the advent of usable, large screen liquid crystal display (LCD) panels, the first truly portable PCs, referred to as laptops, were introduced. These units featured all-in-one, AT-compatible PC boards. The system board included the I/O and video controller functions. Laptops featured built-in keyboards and hinged LCD display panels that flipped up from the case for use. They also used an external power supply and a removable, rechargeable battery. The battery life was minimal and the size was still large enough to be inconvenient at times. However, the LCD viewing screen and external power supply/battery arrangement proved to be useful enough to fuel the portable market. Computer users could easily take work from the office to the home, or to a hotel room while traveling. They could also get work done at traditionally nonproductive times such as on long automobile or airplane rides. An occasional game of computerized cards or golf was always at hand as well.

Additional advancements in IC technology enabled the PC's circuitry to be reduced further so that the unit could achieve sizes of 8.75"d×11"w×2.25"h and beyond. Portables in this size range are referred to as notebook computers. Notebook designers work constantly to decrease the size and power consumption of all the computer's components. Special low-power consumption ICs and disk drives have been developed to extend battery life. The most widely used notebook keyboard is the 84-key version. The keys are slightly smaller and shorter than those found in full-size keyboards. A number of keys or key functions may be combined or deleted from a notebook keyboard.

The continued minimization of the system comes at a cost. Most notably, the number of I/O ports, memory, and disk drive expansion capabilities are limited. In addition, it is not possible to use common, full-sized options adapter cards that are so inexpensive and easy to find.

To overcome the short falls of miniaturization, a wide variety of specialty items aimed at portables have emerged. As mentioned in Chapter 1, small 2-1/2 inch hard disk drives have been developed expressly for the portable market. Other such items include small internal and external modems, special network adapters that plug into parallel printer ports, docking stations (or ports), special carrying cases and brief cases, detachable key pads, clip-on or built-in trackballs, and touch-sensitive mouse pads.

In addition, a sequence of special credit card-like adapter cards has been designed expressly for portable computers. These adapters are standardized through the *Personal Computer Memory Card International Association* (PCMCIA). These cards are more commonly referred to as *PC Cards*. The different types of PCMCIA cards are covered in greater detail in the section “Expansion Slots.”

A `docking port` is a specialized case that enables the entire notebook unit to be inserted into it. Inside, the docking port extends the expansion bus of the notebook. In doing so, it enables the notebook to be connected to a collection of desktop I/O devices, such as full-sized keyboards and CRT monitors, as well as modems and other non-notebook devices.

Even smaller `sub-notebook PCs` have been created by moving the disk drives outside the case and reducing the size of the display screen. Very small sub-notebooks, referred to as `palmtop PCs`, were produced for a short time in the pre-Windows days. These units limited everything as far as possible to reach sizes of 7"w×4"d×1"h. Sub-notebooks have decreased in popularity as notebooks have decreased in weight and cost.

The palmtop market has diminished due to the difficulty of running Windows on such small displays. Human ergonomics also come into play when dealing with smaller notebooks. The screens and keyboards become difficult to see and use as their sizes are decreased.

The drawback of portable computers from a service point of view is that conventions and compatibility disappear. The internal board (or boards) are designed to fit around the nuances of the portable case rather than to match a standard design with standard spacing and connections. Therefore, interchangeability of parts with other machines or makers goes by the wayside. The only source of most portable parts is the original manufacturer. Even the battery case is proprietary. If the battery dies, then you must hope that the original maker has a supply of that particular model. This is true of many of the parts in a portable computer.

Access to the notebook's internal components is usually challenging. Each case design has different methods for assembly and disassembly of the unit. Even the simplest upgrade task can be difficult with a notebook computer. Although adding RAM and options to desktop and tower units is a relatively easy and straightforward process, the same tasks in notebook computers can be difficult. In some notebooks, it is necessary to disassemble the two halves of the case and remove the keyboard to add RAM to the system. In other units, the hinged display unit must be removed to disassemble the unit. After you are inside the notebook you may find several of the components are hidden behind other units.

Table 2.1 compares the features and characteristics of the cases discussed in this section.

Table 2.1

Case comparisons.

Types	DD Bays (Inches)	Expansion Slots	PS (Watts)	Dimensions (Inches)
PC/PCXT	5¼ (4)	6-8	56-135	19×17×5.5
AT	5¼e (2) 5¼i (3)	6-8	150-220	21×17×6.5
Baby AT	5¼e (2) 5¼i (3)	6-8	150-200	19×17×6.5**
Low Profile	5¼ (1-2*) 3½ (1-2)	2-4	150-200	16×15×5
Mini Towers	5¼e(2-3*) 5¼i (2-3)	6-8	150-250	7.5×18×17-24**
Towers	5¼e (4*) 5¼i (1-4)	8-12	150-400	7.5×18×24-30**

*The number of internally (i) and externally (e) accessible drive bays varies from manufacturer to manufacturer. Also, some 3¼"-specific bays may be included in case styles.
**The dimensions of these case styles vary from manufacturer to manufacturer.

Power Supplies

Objective

The system's power supply unit provides electrical power for every component inside the System Unit, and also supplies AC power to the video-display monitor. It converts commercial electrical power received from a 120VAC, 60Hz (or 220VAC, 50Hz outside the U.S.) outlet into other levels required by the components of the system. In desktop and tower PCs, the power supply is the shiny metal box located at the rear of the system unit.

In all, the desktop/tower power supply produces four different levels of efficiently regulated DC voltage. These are: +5V, –5V, +12V, and –12V. It also provides the system's ground. The IC devices on the system board and adapter cards use the +5V level. In particular, the P8/P9 connectors provide the system board and the individual expansion slots with up to 1 ampere of current each. All four voltage levels are available for use at the expansion slot connectors.

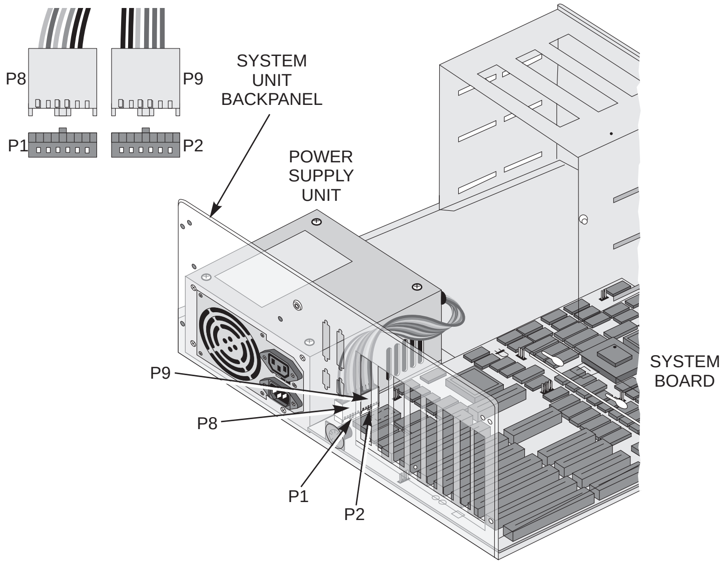
In the event the power supply detects the presence of inadequate voltage levels or damaging `spikes` (power line transients), it has the capability built into it to shut down and not supply power to the system. This feature protects the system's circuitry, but the contents of the RAM memory are lost in such a shutdown.

Several bundles of cable emerge from the power supply to provide power to the components of the system unit and to its peripherals. It delivers power to the system board and its expansion slots through two bundles typically marked P8 and P9. The physical construction of these connectors is significantly different from that of the other bundles. They are designed to be plugged into the system board's P1 and P2 power plugs, respectively, as depicted in Figure 2.13.

The P8/P9 connectors are normally keyed and numbered. However, their construction and appearance are identical. The voltage levels associated with each plug are different and `severe damage could result` to the computer if you reverse them. The power connector labeled P8 should be plugged into the circuit-board connector nearer the rear of the unit, while connector P9 should be plugged into the connector next to it. A good rule of thumb to remember when attaching these two connectors to the system board is that the black wires in each bundle should be next to each other.

Figure 2.13

The P1/P2—P8/ P9 connections.



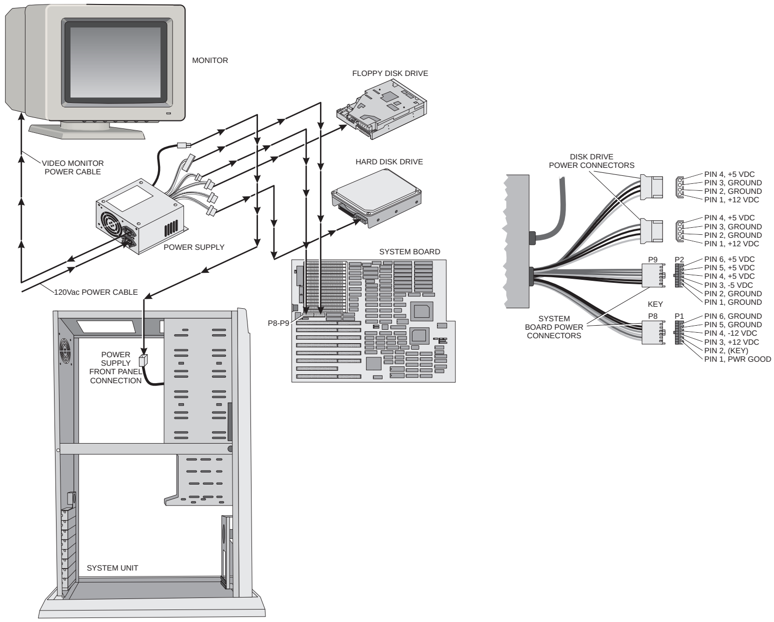
The other power-supply bundles are used to supply power to optional systems, such as the disk and CD-ROM drives. These bundles provide a +5 and +12VDC supply. The +5V supply provides power for electronic components on the optional devices while the +12V is used for disk drive motors and other devices that require a higher voltage.

Power is delivered to the monitor through a special plug in the power supply's back plate. The power supply (as well as the rest of the system) is switched on via the power switch. In older units, the power switch is an integral part of the power supply unit and extends from its right side. In most newer units, the ON/OFF switch is located on the front panel of the system unit and connects to the power supply by a cable. Simply push the button to turn the system ON, and push it again to turn the system OFF. Figure 2.14 illustrates the typical power-supply connections found in a desktop or tower unit.

Within the United States, a grounded, three-prong power cord provides the AC input voltage to the power supply. The smaller vertical blade in the connector is considered the `hot` or `phase` side of the connector. A small slide switch on the back of the unit permits the power supply to be switched over to operate on 220VAC input voltages found outside the United States. When the switch is set to the 220 position, the voltage supplied to the power supply's monitor outlet is also 220. In this position, it is usually necessary to exchange the power cord for one that has a plug suited to the country in which the computer is being used.

Figure 2.14

Power supply connections.



Power supply units come in a variety of shapes and power ratings. The shapes are determined by the type of case in which they are designed to be used. Figure 2.15 illustrates the various power supply shapes. Typical power ratings include 150-, 200-, and 250-watt versions.

Notebooks and other portables use a detachable, rechargeable battery and an external power supply, as illustrated in Figure 2.16. (Battery sizes vary from manufacturer to manufacturer.) They also employ power-saving circuits and ICs designed to lengthen the battery's useful life. The battery unit contains a recharging regulator circuit that enables the battery to recharge while it is being used with the external power supply. As in other hardware aspects of notebook computers, there are no standards for the power supply units. They use different connector types and possess different voltage and current delivery capabilities. Therefore, a power supply from one notebook does not necessarily work with another.

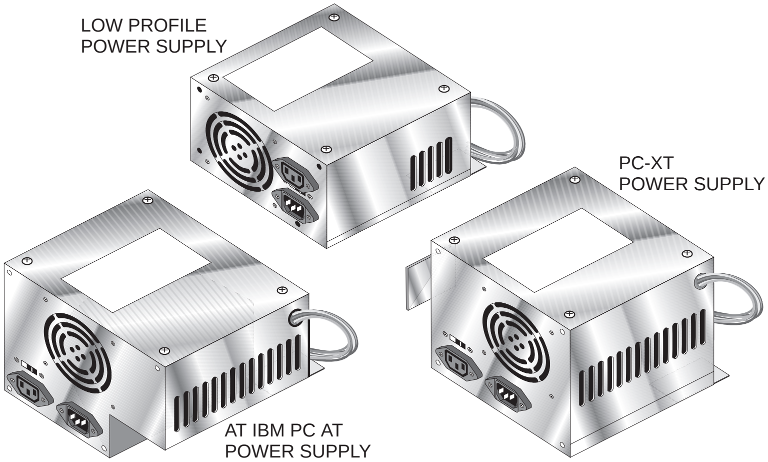


Figure 2.15

Desktop/tower power supplies.

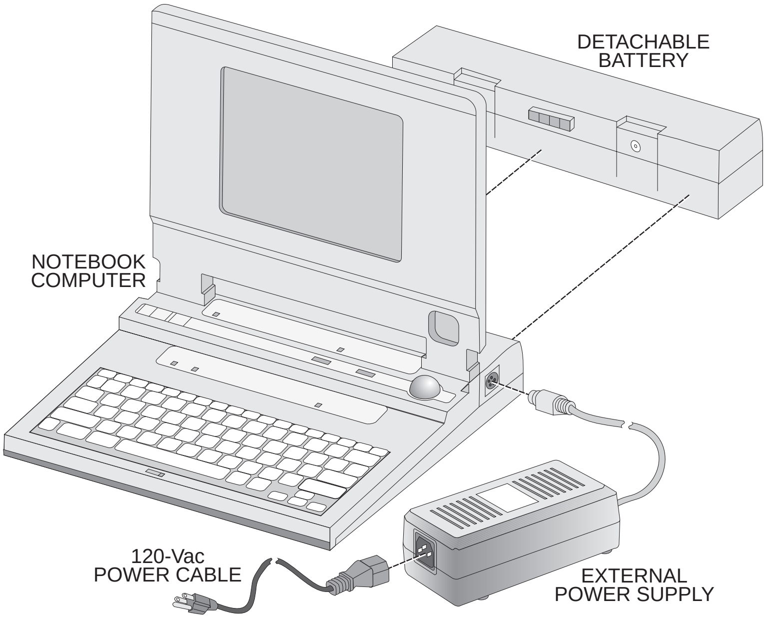


Figure 2.16

Laptop/notebook power supplies.

System Boards

Objective

The system board is the center of the PC-compatible microcomputer system. It contains the circuitry that determines the computing power and speed of the entire system. In particular, it contains the microprocessor and control devices that form the brains of the system. The major components of interest on a PC system board are: the microprocessor, the system's primary Read-Only (ROM), Random Access (RAM), and cache memory sections, expansion slot connectors, and microprocessor support ICs that coordinate the operation of the system. A typical system board layout is depicted in Figure 2.17.

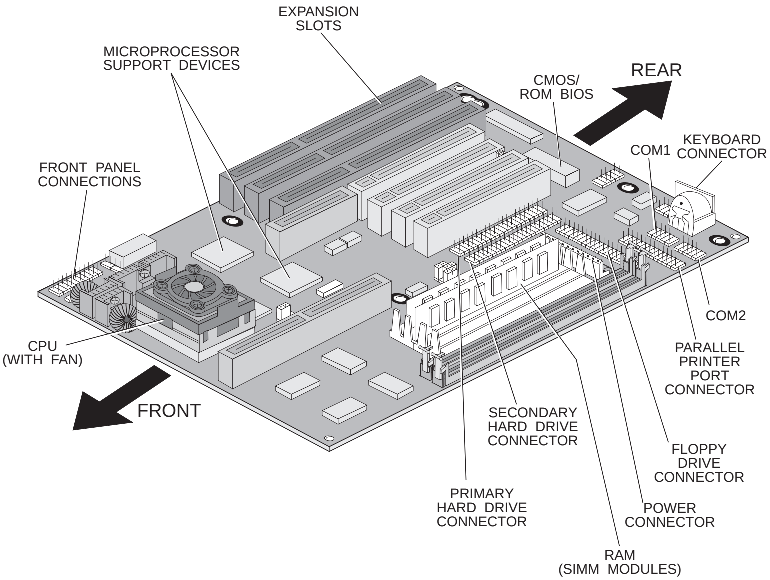


Figure 2.17

The parts of a typical system board.

Major Components

For orientation purposes, the end of the board where the keyboard connector, expansion slots, and power connectors are located is generally referred to as the rear of the board.

The system board communicates with various optional Input/Output (I/O) and memory systems through adapter boards that plug into its `expansion slots`. These connectors are normally located along the left rear portion of the system board so that the external devices they serve can access them through openings at the rear of the case.

Several different types of expansion slots are in use today. A particular system board may contain only one type of slot, or it may have a few of each type of expansion slot. Be aware that adapter cards are compatible with particular types of slots, so it is important to know which type of slot is being used. The major expansion slot types are the following:

- 8-bit PC-Bus
- 16-bit AT or ISA bus
- 32-bit Extended ISA (EISA) and Microchannel Architecture (MCA) buses
- Video Electronics Standards Association (VESA) and Peripheral Component Interconnect (PCI) local buses

These expansion slots are discussed later in this chapter (see the "Expansion Slots" section for more information) and then again in more detail in Chapter 6, "System Boards."

The system board receives power from the power supply unit through a pair of 6-pin power supply connectors that are typically labeled as P1 and P2. These connectors are always located directly beside each other and are keyed so that the power cable cannot be plugged in backwards. However, the P1 and P2 connectors are identical and can be reversed. This condition causes severe damage to system components if power is applied to the system. The P1/P2 connectors are often located along the right rear corner of the system board.

Similarly, the keyboard connector is normally located along the back edge of the board. In most PC-compatible systems, the keyboard connector is a round, 5-pin DIN connector.

Primary Memory

Objective

All computers need a place to temporarily store information while other pieces of information are being processed. As discussed in Chapter 1, "Microcomputer Fundamentals," information storage is normally conducted at two different levels in digital computers: primary memory (made up of semiconductor RAM and ROM chips) and mass storage (usually involving floppy and hard disk drives).

Most of the system's primary memory is located on the system board. Primary memory typically exists in two or three forms on the system board:

- Read-only memory (ROM), which contains the computer's permanent startup programs.
- Random access memory (RAM), which is quick enough to operate directly with the microprocessor and can be read from, and written to, as often as desired.
- Cache memory, which is a fast RAM system specially designed to hold information that the microprocessor is likely to use.

ROM devices store information in a permanent fashion and are used to hold programs and data that do not change. RAM devices retain the information stored in them only as long as electrical power is applied to the IC. Any interruption of power causes the contents of the memory to vanish. This is referred to as volatile memory. ROM, on the other hand, is nonvolatile.

Every system board contains one or two ROM ICs that hold the system's Basic Input/Output System, or BIOS program. The BIOS program contains the basic instructions for communications between the microprocessor and the various input and output devices in the system. Until recently, this information was stored permanently inside the ROM chips, and could be changed only by replacing the chips. As mentioned in Chapter 1, "Microcomputer Fundamentals," advancements in EEPRROM technology have produced Flash ROM devices that enable new BIOS information to be written (downloaded) into the ROM to update it. The download can come from an update disk or another computer. Unlike RAM ICs, the contents of the Flash

ROM remain after the power has been removed from the chip. In either case, the upgraded BIOS must be compatible with the system browser it is being used in and should be the latest version available.

The information in the BIOS represents all the intelligence that the computer has until it can load more information from another source, such as a hard or floppy disk. Taken together, the BIOS software (programming) and hardware (the ROM chip) are referred to as `firmware`. These ICs can be located anywhere on the system board, but they are usually easy to recognize due to their size and immediate proximity to one another.

In older PC designs—XT and AT—the system's RAM memory was comprised of banks of discrete RAM ICs in Dual In-line Pin (DIP) sockets. Most of these system boards arranged 9 pieces of 1 x 256 Kbit DRAM chips in the first two banks (0 and 1) and 9 pieces of 1 x 64 Kbit chips in banks two and three. The two banks of 256K chips provided a total of 512 Kbytes of storage (the ninth chip of each bank supplied a parity bit for error checking). The two banks of 64k chips extended the RAM memory capacity out to the full 640 Kbytes. As with the 256K chips, the ninth bit was for parity.

Some system boards used two 4 x 256 Kbit chips with a 1 x 256 Kbit chip to create each of the first two banks. In any event, the system would typically run with one bank installed, two banks installed, or all of the banks installed. Bank 0 had to be filled first, followed by bank 1 and then all four.

Intermediate clone designs, placed groups of RAM ICs on small 30-pin daughter boards that plugged into the system board vertically. This mounting method required less horizontal board space. These RAM modules had pins along one side of the board and were referred to as *Single In-line Pin* (SIP) modules.

Further refinements of the RAM module produced snap in *Single In-line Memory Modules* (SIMMs) and *Dual In-line Memory Modules* (DIMMs). Like the SIP, the SIMM and DIMM units mount vertically on the system board. However, instead of using a pin and socket arrangement, both use special snap-in sockets that support the module firmly. SIMMs and DIMMs are also keyed, so that they cannot be plugged in backwards. SIMMs are available in 30- and 72-pin versions while DIMMs are larger 168-pin boards.

SIMM and DIMM sockets are quite distinctive in that they are normally arranged side by side. However, they can be located anywhere on the system board. SIMMs typically come in 8- or 32-bit data storage configurations. The 8-bit modules must be arranged in banks to match the data bus size of the system's microprocessor. In order to work effectively with a 32-bit microprocessor, a bank of four 8-bit SIMMs would need to be used. Conversely, a single 32-bit SIMM could do the same job.

DIMMs, on the other hand, typically come in 32- and 64-bit widths to service more powerful microprocessors. Like the SIMMs, they must be arranged properly to fit the size of the system data bus. In both cases, the modules can be accessed in smaller 8- and 16-bit segments. SIMMs and DIMMs also come in 9, 36, and 72-bit versions that include parity checking bits for each byte of storage. PCs are usually sold with less than their full RAM capacity. This allows users to purchase a less expensive computer to fit their individual needs and yet retain the option to install more RAM if future applications call for it. DIP, SIP, SIMM and DIMM modules are depicted in Figure 2.18.

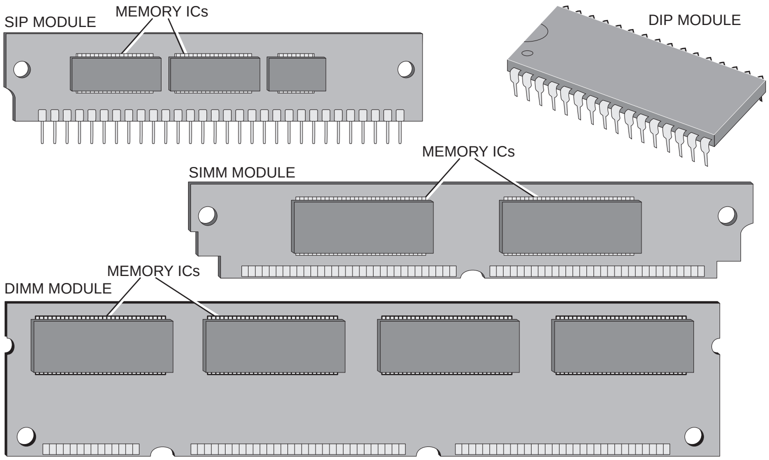


Figure 2.18

DIP, SIP, SIMM, and DIMM memory modules.

SIMM and DIMM sizes are typically specified in an a-by-b format. For example, a 2x32 SIMM specification indicates that it is a dual, nonparity, 32-bit (4-byte) device. In this scheme, the capacity is derived by multiplying the two numbers and then dividing by eight (or 9 for parity chips). DIP, SIP, SIMM, and DIMM modules are depicted in Figure 2.18.

Microprocessors

Objective

The microprocessor is the major component of any system board. It executes software instructions and carries out arithmetic operations for the system. These ICs can take on a number of different package styles depending on their vintage and manufacturer.

The 8088 processor was housed in a 40-pin DIP package. It was used on the original IBM PC and PC-XT units. It featured a 20-bit address bus, an 8-bit data bus, and 16-bit internal word size. The 20-bit address bus enabled it to access 1 megabyte of address space. The mismatch between its internal word size and external data bus size required multiplexed, two-transfer operations with external system devices. Another, more expensive version, called the 8086, featured a full 16-bit external data bus. Although some XTs used 8086-based system boards, the vast majority of PCs, XTs, and clones used the 8088.

The 80286 processors used a variety of 68-pin IC types. These processors were used in the IBM PC-AT and its compatibles. It featured a 24-bit address bus, 16-bit data bus, and 16-bit internal word size. The 24-bit address bus enabled it to directly access up to 16MB of address space, even though DOS could handle only the 1MB that it had been designed to handle with the 8088 systems. The full 16-bit internal and external word size gave it a 4X speed increase over 8088 systems running at the same clock speed. 80286s were produced in various speed ratings.

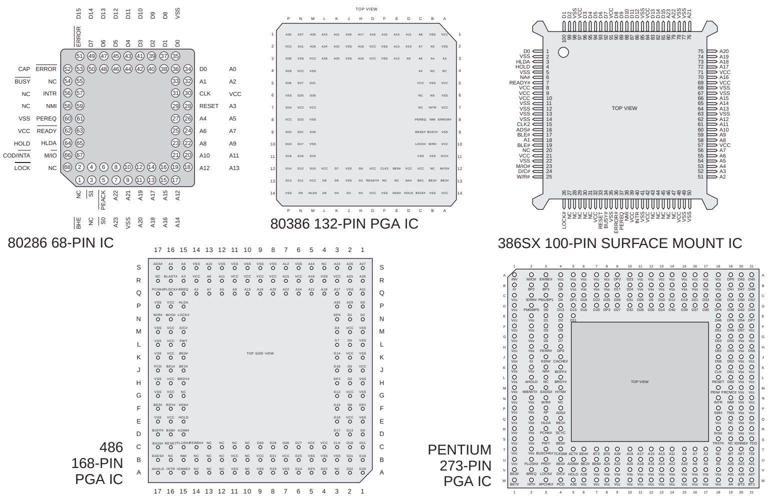
The 80386DX followed with a 132-pin `Pin Grid Array` (PGA) IC package, while a more economical 80386SX was produced in a 100-pin surface-mount IC package. These microprocessors were mainly used in AT clone systems. The DX version provided a 32-bit address bus, 32-bit data bus, and 32-bit internal word size. The 32-bit address bus provided up to 4 gigabytes of memory addressing. The SX version featured a reduced 24-bit address bus and 16-bit data bus. Both versions were produced in a variety of operating speeds and included advanced addressing modes.

The 80486 and Pentium microprocessors returned to 168-pin and 273-pin PGA packages. The 80486 featured a 32-bit address bus, 32-bit external data bus, and 64-bit internal word size. The Pentium also features a 32-bit address bus. However, both the internal and external word size is 64 bits. Both units include onboard math coprocessors for intense numerical operations and special built-in memory areas, called `cache memory`, for high-speed data access to selected data. Like the 80386, these microprocessors are typically used in advanced AT clone computers.

These microprocessor packages are depicted in Figure 2.19.

Figure 2.19.

Microprocessors.



PC Manufacturers mount microprocessors in sockets so that they can be replaced easily. This allows a failed microprocessor to simply be exchanged with a working unit. More often though, the microprocessor is replaced with an improved version to upgrade the speed or performance of the system.

The notches and dots on the various ICs are important keys when replacing a microprocessor. They specify the location of the IC's number 1 pin. This pin must be lined up with the pin-1 notch of the socket for proper insertion. In older systems, the microprocessors had to be forcibly removed from the socket using an IC extractor tool. As the typical microprocessor's pin count increased, special *Zero Insertion Force* (ZIF) sockets were implemented that allowed the microprocessor to be set in the socket without force and then clamped in place. An arm-activated clamping mechanism in the socket shifts to the side, locking the pins in place. All the microprocessors discussed here are covered in greater detail in Chapter 6, along with their significant variations.

For mathematically intensive operations, some programs may shift portions of the work to special high-speed math coprocessors, if they are present in the system. These devices are specialized microprocessors that work in parallel with the main microprocessor, and extend its instruction set to speed up math and logic operations. They basically add large register sets to the microprocessor, along with additional arithmetic processing instructions. The 8088, 80286, 80386SX/DX, and 80486SX microprocessors used external coprocessors. The 80486DX, and Pentium processors, have built-in coprocessors.

Chip Sets

Although the microprocessor is the main IC of the system board, it is certainly not the only IC on the board. Microprocessor manufacturers always produce microprocessor-support chip sets that provide auxiliary services for the microprocessor. The original PC system board used a standard 6-IC 8088 chip set to support the microprocessor. Even so, these early support sets still required a good number of discrete logic ICs to complete the system.

The original AT increased the IC count on the system board by doubling up the DMA and interrupt channel counts. However, when the PC-AT architecture became the acceptable industry standard, several IC manufacturers began developing `very large scale integration` (VLSI) chip sets to perform the standard functions of the AT system board. In the fall of 1985, the company Chips & Technology released a 5-chip chipset that replaced the functions of 63 smaller ICs on the PC-AT system board.

For the IC manufacturer, this meant designing chip sets that would use the same basic memory map that was employed in the IBM PC-AT. (For example the chip set's programmable registers, RAM, ROM, and other addresses had to be identical to those of the AT.) Therefore, instructions and data in the program would be interpreted, processed, and distributed the same way in both systems. In doing so, the supporting chip set decreased from eight major ICs and dozens of `small scale integration` (SSI) devices to two or three VLSI chips and a handful of SSI devices.

In some highly integrated system boards, the only ICs that remain are the microprocessor, one or two ROM BIOS chips, a single chipset IC, and the system's memory modules.

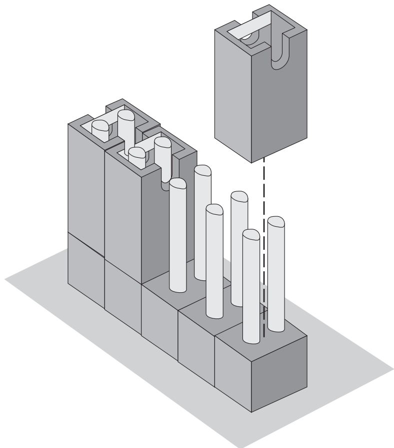
Connectors & Jumpers

System boards possess a number of jumpers and connectors of which you must be aware. PC-compatible system boards include micro switches and jumper blocks (called `BERG connectors`) to select operating options such as speed, video display type (Color/Mono) being used, memory-refreshing Wait-States (special timing periods used by the computer to make certain that everything happens on schedule), installed RAM size, and so forth. You may be required to alter these settings if you change a component or install a new module in the system.

Figure 2.20 illustrates the operation of typical configuration jumpers and switches. A metal clip in the cap of the jumper creates an electrical short between the pins across which it is installed. When the cap is removed, the electrical connection is also removed and an electrically open condition is created.

Figure 2.20

Jumpers and configuration switches.



System boards and several types of I/O cards use micro switches for configuration purposes. These micro switches are normally integrated into a dual in-line pin package, as illustrated in Figure 2.20. The switches may use a rocker or slide switch mechanism to create the short or open condition. The switches are typically numbered sequentially and marked for on/off positioning. Because the switches are so small, they may simply be marked with an ON or OFF, or with a 1 or 0.

It is usually necessary to consult the system board, or adapter's Installation Guide, to locate and properly set configuration jumpers and switches. The Installation Guide typically provides the locations of all the board's configuration jumpers and switches. It also defines the possible configuration settings, along with corresponding switch or jumper positions.

The system board is connected to the front panel's indicators and controls by BERG connectors. Over time, these connection points have become fairly standard between cases. The normal connections are the `Power LED`, `Turbo LED`, `Turbo Switch`, `Keylock Switch`, `Reset Switch`, and `System Speaker`. A typical front panel connector layout is described in Figure 2.21. It is necessary to access these points when the system board is replaced or upgraded. Additional system board connectors that must be dealt with include the keyboard and power supply connectors.

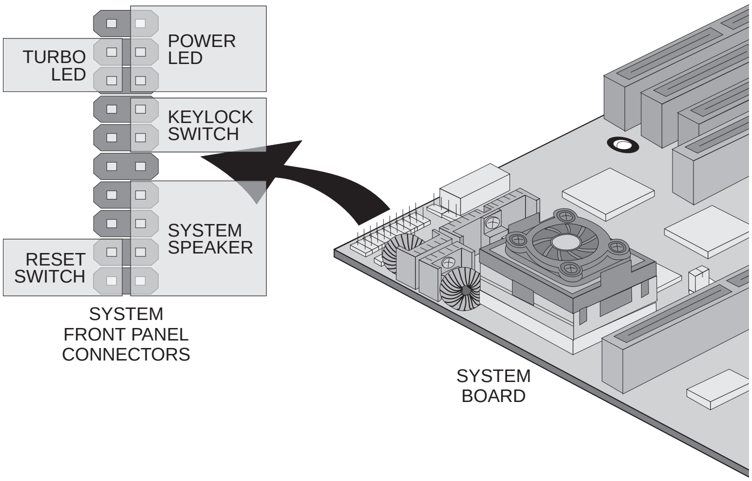


Figure 2.21

System board connection points.

Configuration Settings

Each time the system is turned ON, or RESET, the BIOS program checks the system's `Configuration Settings` to determine what types of optional devices may be included in the system. Depending upon the model of the computer, the configuration information may be read from hardware jumper or switch settings, from battery-powered RAM, or, in some cases, a combination of jumper and software settings. The PC, PC-XT and their clones used hardware switches for configuration purposes. When IBM introduced the PC-AT, it featured a battery-powered RAM area that held some of the system's advanced configuration information. This storage area was referred to as CMOS RAM. With that unit, configuration was performed from a floppy disk diagnostic program. Clone BIOS manufacturers quickly added the advanced configuration function to their BIOS chips. Clone system boards also added a rechargeable, Ni-Cad battery to the board to maintain the information when the system was turned off. Newer systems, do not use a rechargeable Ni-Cad battery for the CMOS storage. Instead, the CMOS storage area has been integrated with a 10-year, nonreplaceable lithium cell in a single IC package.

Because these settings are the system's only way of getting information about what options are installed, they must be set to accurately reflect the actual options being used with the system. If not, an ERROR occurs. You should always suspect configuration problems if a machine fails to operate immediately after a new component has been installed. These CMOS configuration values can be accessed for change by pressing the CTRL and DEL keys (or some other key combination) simultaneously during the bootup procedure.

Newer microcomputers possess the capability to automatically reconfigure themselves for new options that are installed. This feature is referred to as `Plug-and-Play` (PnP) capability. In 1994, Microsoft and Intel teamed up to produce a set of system specifications that would enable options added to the system to be configured automatically for operation. Under this scenario, the user would not be involved in setting hardware jumpers or CMOS entries. To accomplish this, the system's BIOS, expansion slots, and adapter cards are designed in a manner so that they can be reconfigured automatically by the system software.

During the startup process, the PnP BIOS looks through the system for installed devices. Devices designed for PnP compatibility can tell the BIOS what types of devices they are and how to communicate with them. This information is stored in memory so that the system can work with the device. PnP information is scattered throughout the remainder of the text of this chapter as it applies to the topic being covered.

Expansion Slots

In the back-left corner of most system boards are up to eight expansion slots. These slots enable the system's peripheral devices to be connected to the system board. Optional I/O devices, or their PC-compatible interface boards, may be plugged into these slots to connect the device to the system unit's address, data, and control buses.

8-Bit Slots

The expansion slots in the original PC, PC-XT, and their compatibles contained an 8-bit, bidirectional data bus and 20 address lines. They also provided six interrupt channels, control signals for memory and I/O read or write operations, clock and timing signals, and three channels of DMA control lines. In addition, the bus offered memory refresh timing signals, and an I/O channel check line for peripheral problems, as well as power and ground lines for the adapters that plug into the bus. I/O devices attached to the bus had to be addressed using the system's I/O-mapped address space.

This expansion slot configuration became a `de facto` connection standard in the industry for 8-bit systems. It was dubbed the `PC-bus` standard. A de facto standard is one that becomes established through popular use rather than through an official certifying organization. Figure 2.22 shows how the PC-bus's 62 lines are arranged at the expansion slot's connector.

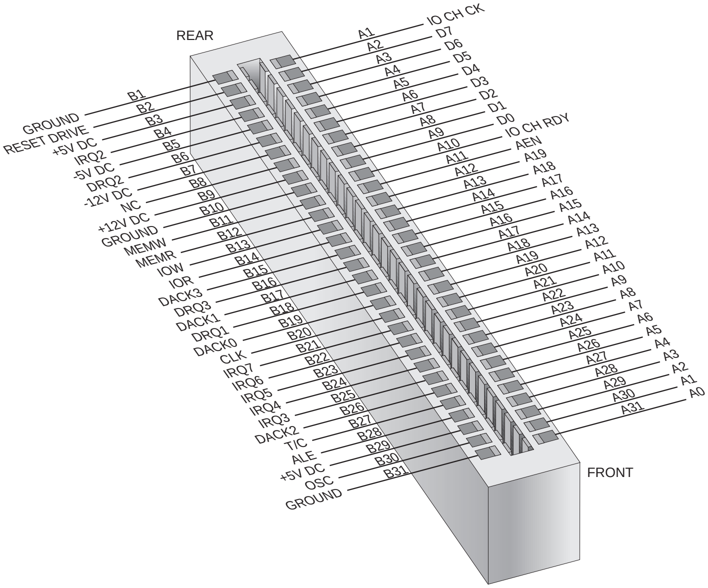
16-Bit Slots

The overwhelming popularity of the IBM-PC AT established it as the 16-bit standard to which all other PC-compatible equipment is compared. Its 16-bit expansion bus specification became the industry standard for all 16-bit systems and devices. The 16-bit data bus enabled twice as much data to pass back and forth through the connector in a given amount of time. This made transfers with 16-bit microprocessors smooth. No high/low-byte bus multiplexing is required.

The ISA bus included twice as many interrupt and DMA channels as the PC Bus specification. This made it possible to connect more peripheral devices to these systems. In order to maintain compatibility with older adapter cards, the transfer speed for the ISA bus was limited to the same speed as that of the older PC Bus. Figure 2.23 describes an ISA-compatible expansion slot connector. These expansion slots actually exist in two parts: the slightly altered, 62-pin I/O connector, similar to the standard PC-bus connector, and a 36-pin auxiliary connector.

Figure 2.22

An 8-bit PC bus expansion slot.



32-Bit Slots

Following the development of fast, 32-bit microprocessors, it was normal for designers to search for a new bus system to take advantage of the 32-bit bus and the higher speed of operation. IBM introduced a new line of Personal System/2 (PS/2) computers, that featured a new expansion bus standard that it called `Micro Channel Architecture` (MCA). Although similar to the ISA edge connector in appearance, the MCA 16-bit connector is physically much smaller than that of the 16-bit ISA bus connector. (The MCA's edge connector is 2.8 inches long, compared to the 5.29-inch length of the ISA bus.) The contacts along the edge connector are separated by only 0.05 inch. This is only half the distance allotted between contacts in the ISA connector. The smaller size is also seen in the dimensions of MCA-compatible adapter cards. Both types of edge connectors are depicted in Figure 2.24.

Figure 2.23

A 16-bit ISA expansion slot.

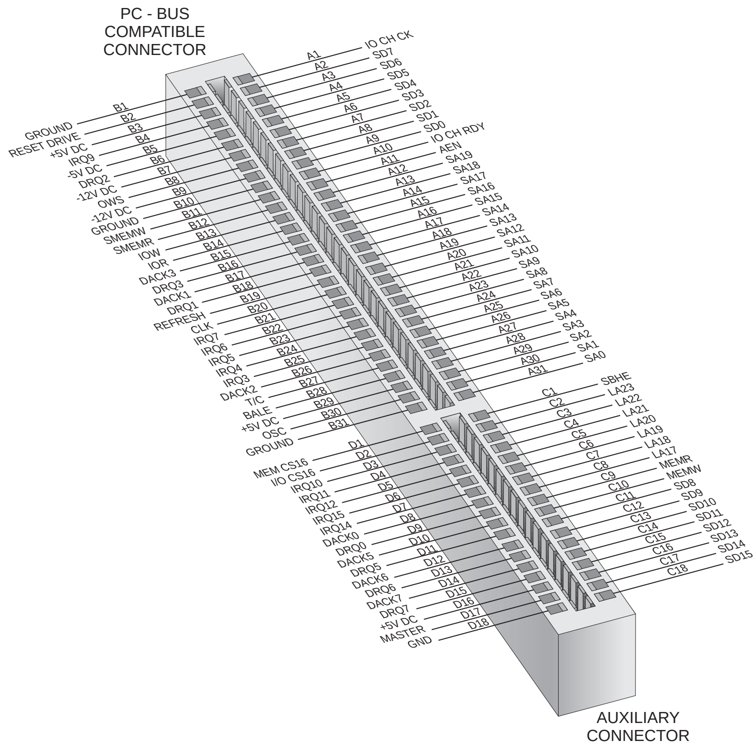
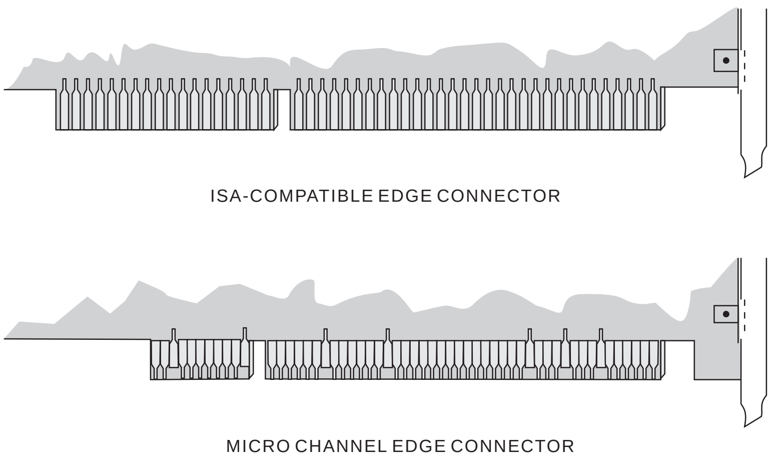


Figure 2.24.

32-bit MCA and 16-bit ISA edge connectors.



In an attempt to develop an acceptable, 32-bit I/O bus standard that would remain compatible with the large quantities of hardware and software already designed for ISA-compatible computers, a group of computer hardware and software manufacturers (known as `the gang of nine`) banded together and introduced specifications for an extension to the ISA bus. Compaq, AST Research, Epson, Hewlett-Packard, Olivetti, NEC, Tandy, Wyse, and Zenith Data Systems pooled their efforts to extend the ISA world, and christened the `EISA Bus`.

The key to the EISA bus specification is found in the design of its expansion slot connector. The connector is designed so that it can accept traditional 8- and 16-bit ISA cards, as well as the newer 32-bit EISA cards. This flexibility is achieved by incorporating a two-level approach to the connector.

Signals associated with the standard ISA connection are terminated in the upper row of contacts while new EISA signals are interleaved between the ISA signals on the lower row of contacts. In all, the EISA standard adds 55 new signal lines to the bus along with 35 additional power supply and ground contacts.

Small raised keys in the bottom of the connector prevent ISA-style cards from being inserted into the full depth of the connector. Therefore, the contacts on the edge connector of the ISA card can contact the standard ISA signals only along the top of the expansion slot connector. However, the edge connectors of EISA cards, which are much deeper than ISA cards (0.52 inch to 0.31 inch), have notches that match the keys in the expansion slot connector and enable the EISA card to be fully inserted. In this way, both rows of contacts make connection when an EISA expansion card is inserted into an EISA expansion slot. It should be apparent that this connection scheme does not work in reverse. An EISA card does not work in a standard ISA slot connector. Although this action should not cause damage to occur, it renders the host computer inoperative until the card is removed.

Both the EISA and MCA architectures were designed with Plug-and-Play capabilities in mind. The EISA bus supports PnP in only a hardware fashion, while MCA supports PnP in both hardware and software manners. All the adapters used in the MCA architecture possess the capability to automatically identify themselves to the system and be reconfigured by the system for optimum performance. The system is able to obtain information from the adapter cards as to what type they are, where they are located, and to which resources they need access. With this information, the system can check the other adapters and onboard intelligent devices, and then reconfigure all the devices to work together.

Local Bus Slots

In recent years, different manufacturers have implemented proprietary expansion bus designs to improve their system boards. These designs increase the speed and bandwidth between the microprocessor and a few selected peripherals by creating a special bus between them. These special bus designs are called `local buses`. The local bus connects special peripherals to the system board (and the microprocessor) through a proprietary expansion slot connector and enables the peripheral to operate at speeds close to the speed of the microprocessor.

When these designs began to appear, the peripherals that could be used in the proprietary slots were available only from the original system board manufacturer. The industry soon realized the benefits of such designs and the need for some standards. Currently, two local bus designs have gained enough acceptance to have third-party manufacturers design products for them. These designs are the VESA and PCI local bus specifications. Most Pentium system boards include a combination of ISA and PCI expansion slots.

The VESA local bus was developed by the `Video Electronics Standards Association`. This local bus specification, also referred to as the VL-Bus, was originally developed to provide a local bus connection to a video adapter. However, its operation has since been defined for use by other adapter types, such as drive controllers, network interfaces, and other hardware. The VESA connector is depicted in Figure 2.25.

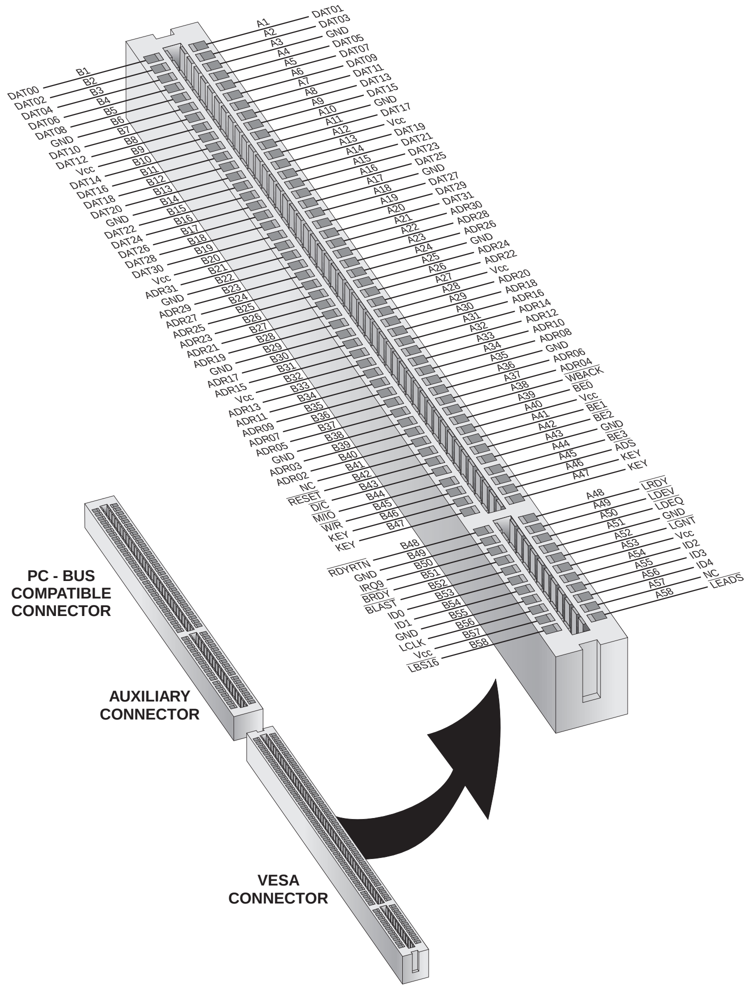
The VL-Bus defines a local bus that can operate at up to 66MHz and is designed for use with 386 or 486 microprocessors. However, newer revisions multiplex the address and data buses to provide a 64-bit data bus for use with the Pentium and future generation microprocessors. This newer revision enables a 32-bit adapter to operate in a 64-bit slot, or vice-versa.

The `Peripheral Component Interconnect` (PCI) local bus was developed jointly by IBM, Intel, DEC, NCR, and Compaq. This bus design incorporates three elements: a low-cost, high-performance local bus, an automatic configuration of installed expansion cards, and the capability to expand with the introduction of new microprocessors and peripherals. The data transfer performance of the PCI local bus is 132MBps using a 32-bit bus and 264MBps using a 64-

bit bus. This is accomplished even though the bus has a maximum clock frequency of 33MHz.

Figure 2.25

The VESA local bus slot.



The PCI peripheral device has 256 bytes of onboard memory to hold information as to what type of device it is. The peripheral device can be classified as a controller for a mass-storage device, a network interface, a display, or other hardware. The configuration space also contains control, status, and latency timer values. The latency timer register on the device determines the length of time that the device can control the bus for bus mastering operations. Figure 2.26 illustrates the PCI Local Bus connector.

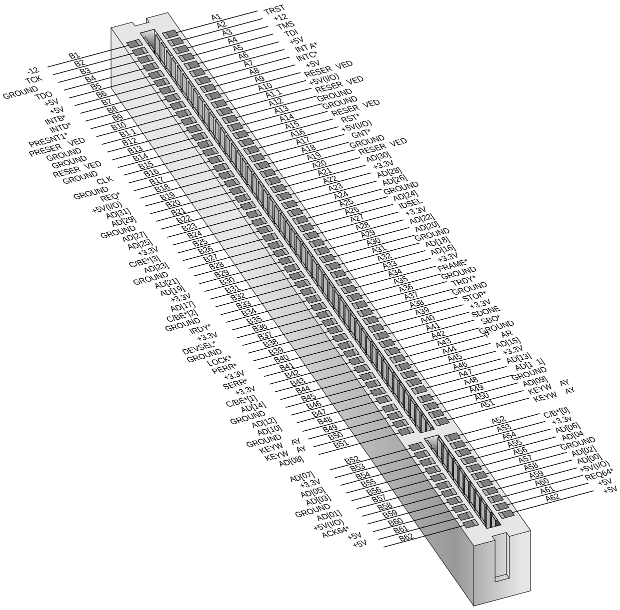


Figure 2.26

The PCI local bus slot.

Both local bus specifications include slot addressing capabilities and reserve memory space to accommodate reconfiguration of each device installed in the system. Unfortunately, system boards that use these expansion slots normally have a few ISA-compatible slots also. This feature seriously disrupts the Plug-and-Play concept because no identification of reconfiguration capabilities was designed into the ISA bus specification.

Table 2.2 compares the capabilities of the various bus types commonly found in personal computers. It is quite apparent that the data transfer rates possible with each new version increases dramatically. The reason this is significant is that the expansion bus is a speed-limiting factor for many of the system's operations. Every access of a disk drive, I/O port, video display, and optional system must pass through this bottleneck. When the bus is accessed, the entire computer must slow down to its operating speed.

Table 2.2

Expansion bus specifications.

Bus Type Channels	Transfer Rate	Data Bits	Address Bits	DMA Channels	INT
PC Bus	1MB/s	8	20	4	6

ISA	8MB/s	16	24	8	11
EISA	32MB/s	32	32	8	11
MCA	20-40MB/s	32	32	xx	-11
VESA	150/275MB/s	32/64	32	xx	1
PCI 2	132/264MB/s	32/64	32	xx	3
PCI 2.1	264/528MB/s	32/64	32	xx	3

PCMCIA Buses

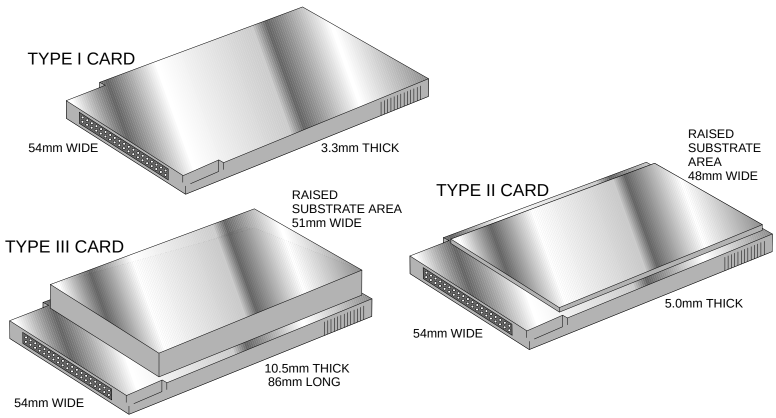
Objective

The PCMCIA bus was developed to accommodate the space-continuous notebook and sub-notebook computer market. Three types of PCMCIA adapters exist. The PCMCIA Type I cards, introduced in 1990, are 3.3mm thick and work as memory expansion units. In 1991, the PCMCIA Type II cards were introduced. They are 5mm thick and support virtually any traditional expansion function, except removable hard drive units. Type II slots are backward-compatible so that Type I cards work in them. Currently, Type III PCMCIA cards are being produced. These cards are 10.5mm thick and are intended primarily for use with removable hard drives. Both Type I and Type II cards can be used in a Type III slot.

All three card types adhere to a form factor of 2.12"w×3.37"l and use a 68-pin, slide-in socket arrangement. They can be used with 8- or 16-bit data bus machines and operate on +5V or +3.3V supplies. The design of the cards enables them to be installed in the computer while it is turned on and running. Figure 2.27 shows the three types of PCMCIA cards.

Figure 2.27

PCMCIA Cards.



Adapter Cards

The openness of the IBM PC, XT, and AT architectures, coupled with their overwhelming popularity, led companies to develop a wide assortment of expansion devices for them.

The adapter cards in the original IBM PC were 13.2"l×4.2"h. The PC normally came with a disk drive adapter card and a video card. These units were referred to as `full-size adapter cards`. They were so long that plastic guide rails at the front of the system unit were present to keep them from flexing due to system heating or transportation. A smaller (6"l×4.2"h) printer adapter card was also made available for the PC. This size card is referred to as a `half-size card`. When the AT appeared, the I/O cards became more powerful and taller (13.2"l×4.8"h).

The AT cards became the standard against which later I/O cards have been measured. As system boards have, adapter cards have developed into smaller, more powerful units. Most current adapter cards are `2/3-Size Cards`, half-size, or smaller cards. In addition, the height of the cards has been significantly reduced. Many adapters are only half the height, or less, of the original AT cards. Therefore, they are referred to as `half-height cards`.

The only real requirements for adapter cards now are that they fit securely in the expansion slot, cover the slot opening in the rear of the system unit, and provide standard connectors for the types of devices they serve. Various adapter card designs are depicted in Figure 2.28.

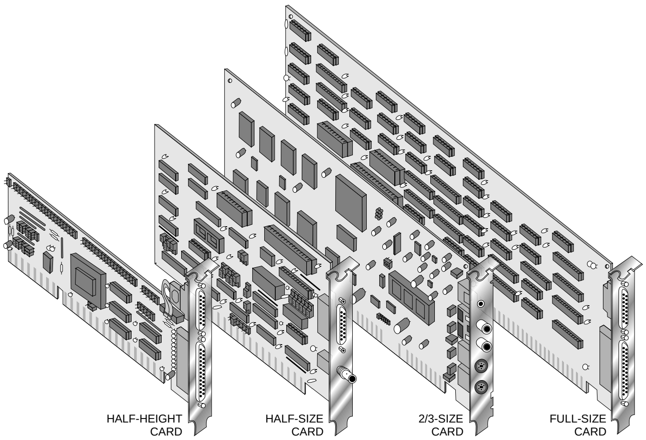


Figure 2.28

Adapter card designs.

Most of the adapter cards that have been developed use hardware jumpers or configuration switches to enable themselves to be configured for the system in which they are to be used. The user sets up the card for operation and solves any interrupt or memory addressing conflicts that occur. These cards are referred to as legacy cards.

In newer PnP systems, adapter cards should have the capability to identify themselves to the system during the startup process, as well as to supply it information about what type of device they are, how they are configured, and to which resources they need access. In addition, these cards must be able to be reconfigured by the system software if a conflict is detected between it and another system device.

Prior to the Pentium-based system boards, two types of adapter cards were traditionally supplied as standard equipment in most desktop and tower PC systems. These were a video adapter Card and a `Multi-I/O` adapter card. However, in Pentium units, the MI/O functions have been built into the system board. Similarly, both the video and I/O functions are an integral part of the system board in portable systems.

Video Adapter Cards

Objective

The video adapter card provides the interface between the system board and the display monitor. The original IBM PCs and XTs offered two types of display adapters, a monochrome (single color) display adapter (MDA) and a Color Graphic Adapter (CGA). Both these units also included the system's first parallel printer port connector.

These initial units have been followed by a number of improved and enhanced video adapters and monitors. The most common type of video adapter card currently in use is the Video Graphic Adapter (VGA) card, as depicted in Figure 2.29. The system uses it to control video output operations.

Every aspect of the computer described so far has dealt with digital signals and circuitry. However, the VGA video standard uses analog signals and circuitry. The main component of most video adapter cards is an ASIC called the `Integrated Video Controller IC`. It is a microprocessor-like chip that oversees the operation of the entire adapter. It is capable of accessing RAM and ROM memory units on the card. The video RAM chips hold the information that is to be displayed on the screen. Its size determines the card's video and color capacities.

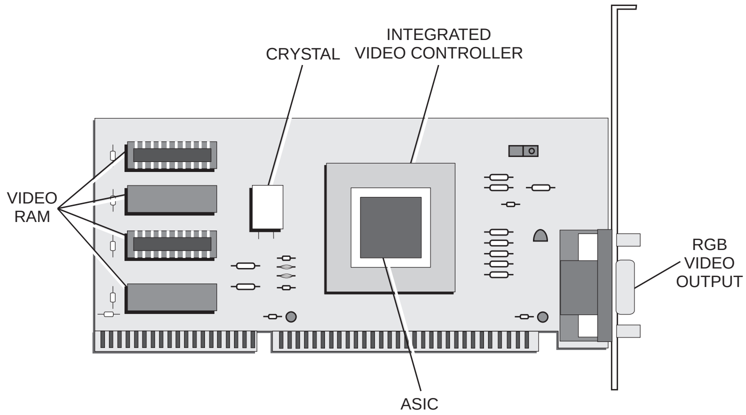


Figure 2.29

A typical VGA card.

The adapter also has a `video BIOS ROM` that is similar to the ROM BIOS on the system board. It is used to store firmware routines that are specific to only video functions. The video controller also contains the `video DAC` (digital-to-analog converter) that converts digital data in the controller into the analog signal used to drive the display. The video output connector is a DB-15 female connector used with analog VGA displays. Unlike earlier video cards, the VGA card does not normally include a parallel printer port.

Multi-I/O Adapter Cards

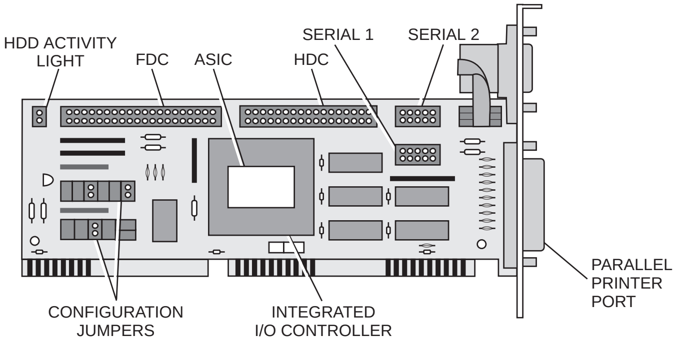
Objective

A Multi-I/O (MI/O) adapter card integrates common I/O functions to provide an array of interfaces for the system. Most MI/O cards combine a `floppy-disk drive controller` (FDC), a `hard-disk drive controller/interface` (HDC), a `game port`, a `parallel printer port`, and two `serial ports` all on one board. Before the advent of VLSI technology, the discrete circuitry involved in most of these functions required a separate adapter card for each function. However, the MI/O adapter combines the basic circuitry for all of these functions into a single ASIC VLSI chip.

Figure 2.30 depicts an MI/O card showing a sample of the IC placement and the location of connectors and configuration jumpers. The disk drives (hard and floppy) are mass storage devices, capable of storing large amounts of data, and are presented in the following section. The game port is used for input of resistive game devices, such as joysticks. The card also includes a programmable parallel printer port, which enables a wide range of printers to be connected to the system. The last function of the adapter is the RS-232C serial interface port. This port supports serial/asynchronous communications for mice and other serial I/O devices, such as modems.

Figure 2.30

A typical MI/O card.



The FDC portion of the adapter is normally capable of controlling two floppy-disk drives (A and B), which connect to the adapter through a 34-conductor flat ribbon cable. Both disk drives can be connected to the cable at once. The system differentiates between the two drives by assigning them `logical drive specifiers` "A:" and "B:" which depend on where they are connected to the cable. The disk drive that is connected to the end of the floppy drive signal cable farthest away from the MI/O card is designated as drive A. When another disk drive is added to the system, it should be plugged into the connector in the middle of the cable. The system then sees it as drive B.

Caution should be used when connecting the disk drives to the adapter card. Pin #1 of the connector and pin #1 of the adapter must be connected to each other. The connecting signal cable has a stripe on one edge to mark pin #1. Ensure that these markings are lined up together when connecting the cable to the adapter card. This is also true of the signal cable at the disk drive end. The relationship between pin #1 of the disk drive, the controller card, and the signal cable is described in Figure 2.31.

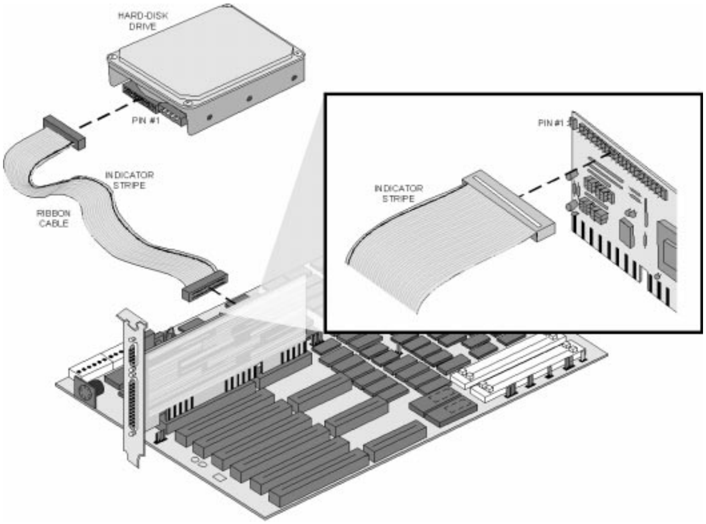


Figure 2.31

Aligning pin #1.

The HDC portion of the adapter is usually capable of controlling two hard-disk drives. The first hard drive is designated as drive C and the second is drive D. The hard drives are connected to the MI/O through a ribbon cable. The hard drives are connected to the controller in much the same manner as the floppy drives are. The system's first logical hard drive is connected to the end of the cable farthest away from the adapter. Observe the same cable orientation that was used for connecting the floppy-disk drives when connecting the cable to the MI/O adapter for the hard drives.

Parallel I/O devices plug into the DB-25 female connector. The MI/O card's serial port connectors may be located on a separate expansion slot cover, which is normally located in the slot on either side of the adapter card. At the top of the featured slot cover is a DB-9 male connector where the mouse can be connected. Beneath the 9-pin connector is a DB-25 male connector that serves as the second serial I/O port.

Game port devices connect to the MI/O card through an external 15-pin DB-15 female connector at the rear of the card. The pins of this 15-pin D-Shell are arranged differently from those in the 15-pin VGA connector. In the VGA connector, the pins are arranged in three rows, while the pins in the Game Port are arranged in two rows.

Some external connections may be made by using a short ribbon cable to connect an external D-shell connector to a BERG strip connector located on the board. With a Y-cable plugged into the 15-pin connector, the game port can support two joystick-like devices.

Several configuration jumpers are normally on MI/O adapters. They are used to set the operating characteristics of the different functions on the card. Jumper blocks are also used to either enable or disable the FDC, HDC, game port, and printer port. Other jumpers are used to assign from which parallel port the printer operates.

Other Adapter Cards

It is safe to say that adapter cards have been designed to add a wide number of different functions to the basic computer system. All the adapter cards depicted so far have adhered to the PC-Bus and ISA standards for connectors. The various types of adapter cards discussed so far are also available in other expansion slot styles.

Some cards may contain a controller or interface for a single function, such as a simple parallel port. Conversely, the adapter card may hold a complete, complex peripheral system, such as a modem. In any event, with the correct interface and control circuitry, virtually anything can be connected to the computer system. Some PC board manufacturers have even developed special prototype adapter cards that have no components on them. This enables hobbyists and designers to create specialty adapter cards for their own applications.

Disk Drives

Objective

The system unit normally comes from the manufacturer with both a floppy-disk drive unit and a hard-disk drive unit installed.

However, the system's disk drive capacity is not usually limited to the standard units installed. In most cases, the system cabinet is designed to hold additional disk drive units. These units can be either 3.5- or 5.25-inch floppy-disk drives, or hard disk drives, or a combination of both. Although three FDD units could physically be installed in most systems, the typical floppy drive controller supports only two floppy-disk drives. Conversely, some systems are smaller and have room for only two disk drive units.

One or more hard-disk drive units can be installed in the system unit, along with the floppy drive(s). The system should normally be set up to recognize a single hard-disk unit in the system as Drive C:. However, a single, physical hard-disk drive can be partitioned into two or more volumes that the system recognizes as logical Drives C:, D:, and so on.

Floppy-Disk Drives

The original standard floppy-disk drive for 8088-based machines was the 5.25-inch, full-height and half-height drive. These drives used disks capable of storing 368,640 bytes (referred to as 360KB) of information. The term `half-height` was used to describe drive units that were half as tall as the `full-height` drive units used with the original IBM PC. Smaller 3.5-inch half-height drives, with disks capable of storing 720KB (737,280 bytes) of information, were also used with 8088-based computers.

The PC-AT and its compatibles originally used advanced high-density 5.25-inch disks that could hold over 1,200,000 bytes (1.2MB) of information.

In advanced machines, such as PS/2 computers and later AT compatibles, high-density, 3.5-inch floppy drives, with disks capable of holding 1.44MB (1,474,560 bytes) are common.

Floppy Disks

Floppy disks come in 5.25- and 3.5-inch diameters. The disks are mylar disks that have been coated with a ferro-magnetic material. They are encased in protective, semi-rigid envelopes that contain low-friction liners that remove dust and contaminants from the disk as it turns within the envelope. Typical floppy drives turn the disk at 300 or 360 RPM and the drive's R/W heads ride directly on the disk surface. Information is written to or read from the disk as it spins inside the envelope. The small LED on the front of the disk drive unit lights up whenever either of these operations is in progress.

Current PC systems use a type of floppy disk referred to as `double-sided, high density` (DS-HD). This means that the disk can be used on both sides, and that advanced magnetic recording techniques may be used to effectively double or triple the storage capacity available with earlier recording techniques. These disks can hold 1.44MB of information by using the `Disk Operating System` (or DOS) software.

This type of floppy disk drive can also operate with an earlier type of floppy disk that is referred to as a `Double-Sided, Double-Density` (DS-DD) disk. This notation indicates that the disks are constructed so that they can be used on both sides, and that they can support recording techniques which doubled the storage capacity available with earlier recording techniques. The earlier disks were referred to as `single-sided` (SS) or `single-density` (SD) disks.

A newer standard for 5.25-inch disks enables up to three times as much data (1.2MB) to be stored on the disk as was possible with the DS-DD disks. These are double-sided, high-density (or simply high-density) disks. These disks can usually be distinguished from the other 5.25-inch disk type by the absence of a reinforcing ring around the drive spindle opening.

Hard-Disk Drives

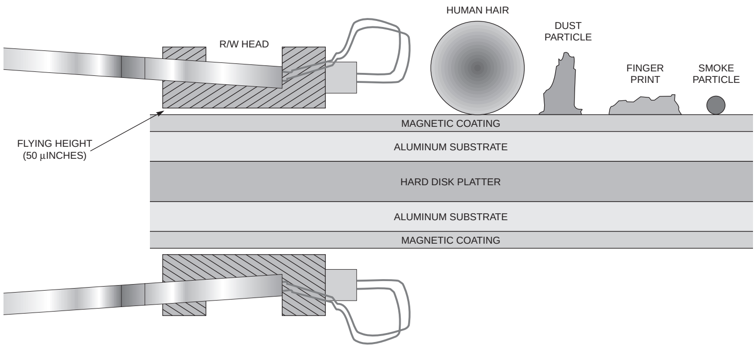
The hard-disk drives normally used with personal computers are referred to as `Winchester hard-disk drive units`. Both full- and half-height versions are popular in desktop and tower systems. These drives typically contain between one and five disks that are permanently mounted inside a sealed enclosure with the R/W head mechanisms. There is one R/W head for each disk surface. The platters are typically turned at a speed of 5,400 RPM. This high rotational speed creates a thin cushion of air around the disk surface that causes the R/W heads to fly just above the disk.

The flying R/W heads glide over the disks at a height of approximately 50 micro inches. This may seem to be an unimportant measurement until you consider the size of a common dust particle or a human hair.

This relationship is illustrated in Figure 2.32. If the R/W head should strike one of these contaminants as the disk spins at high speed, the head would be lofted into the air and then crash into the disk surface. This action could damage the R/W head or the disk surface. This is known as a head crash. To avoid this, hard disks are encased in the sealed protective housing. It is important to realize that at no time should the disk housing be opened to the atmosphere. Repairs to hard-disk drives are performed in special repair facilities having ultra-clean rooms. In these rooms, even particles the size of those in the figure have been removed from the air.

Figure 2.32

Flying R/W heads.



The rigid structure of the hard disk enables its tracks to be placed close together. This in turn makes its storage capacity very high. Typical hard disks may have between 315 and 1,024 sets of tracks (or `cylinders`) that are divided into between 17 and 50 sectors, depending on the diameter of the disk. Sectors generally contain either 256 or 512 bytes. The high speed at which the hard disk revolves also provides very rapid data transfer rates. On a typical 100MB, 3.5-inch hard disk, there are 1,002 cylinders (tracks per side), divided into 32 sectors.

System Speakers

The system's primary audio output device is a 2.25-inch, 8-ohm, 1/2-watt speaker, similar to the one depicted in Figure 2.33. This unit can be located behind the vertical vents in the front panel or under the power supply unit in a small plastic retainer. The system uses the speaker to prompt the user during certain events, and to indicate certain errors in the system, such as video display failures, which can't be displayed on the screen. The speaker can also be controlled by the user through software. Its output frequency range extends through the complete audio range and, with proper programming, can be used to create arcade sounds and music.

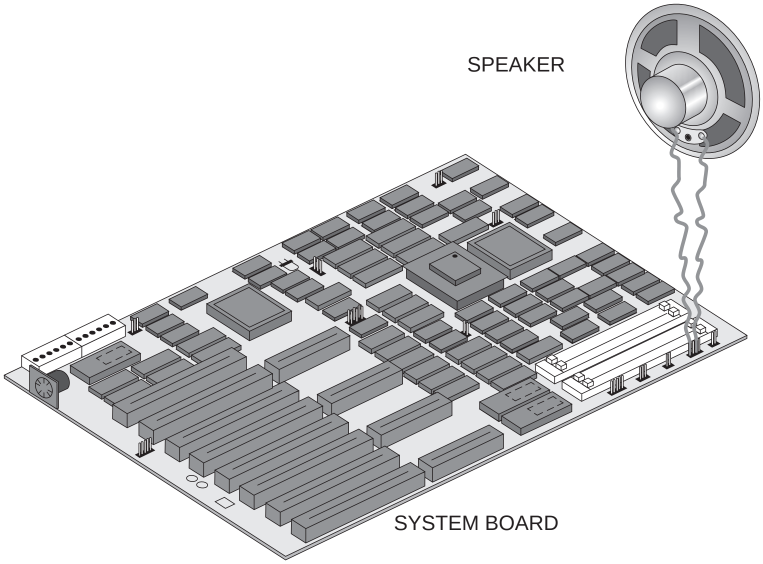


Figure 2.33

The system speaker.

Peripherals

As discussed in Chapter 1, the standard peripherals associated with a personal computer are the keyboard and the CRT monitor. With the rapid growth of GUI-oriented software, the mouse has become a common input peripheral as well. The next most common peripheral has to be the `character printer.` These peripherals are used to produce hard copy output on paper. Besides these common devices, all types of peripheral equipment are routinely added to the computer. As long as there are open expansion slots, or other standard I/O connectors, it is possible to add compatible devices to the system.

The standard input device for PC-compatible computers is the alphanumeric keyboard. Most PCs use a detachable keyboard that is connected to the system by a 6-foot coiled cable. This cable plugs into a round 5-pin DIN connector located on the rear of the system board. The connector is keyed so that it cannot be misaligned.

The most widely used display device for current PCs is the `video graphics array` (VGA) color monitor. The monitor's signal cable connects to a 15-pin D-shell connector at the back of the system unit. A power cable supplies 120VAC power to the monitor from a conventional power outlet or from a special connector on the back of the power supply unit.

The mouse is usually connected to a 9- or 25-pin male D-shell connector on the back panel of the system.

Character printers are normally attached to the system through 25-pin female parallel, or 25-pin male serial D-shell connectors at the back of the unit.

External Connections and Devices

Objective

The system's peripheral devices typically connect to the back of the system unit. Figure 2.34 shows the external connections for the basic system configuration.

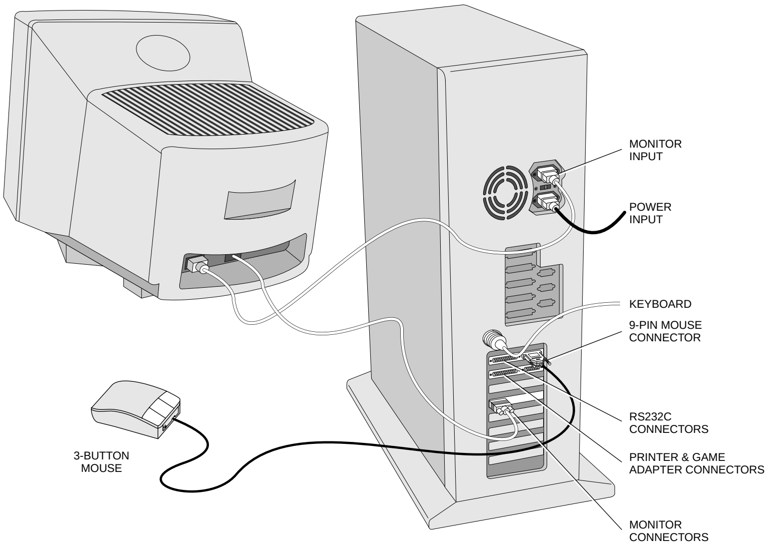


Figure 2.34

External connections.

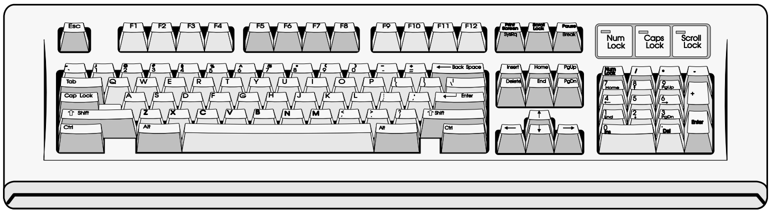
At one side of the system unit is a female power receptacle, which may be used to provide power to PC-compatible monitors. Next to the monitor power receptacle is the power supply's input power connector. Beside the power connector is the power supply's fan vent. In a small opening near the power supply openings is the circular, 5-pin DIN connector for connecting a keyboard to the system. Across the remainder of the back plate are the eight option adapters slot openings. Moving across the panel, you first encounter the system's two RS-232C connectors. In the next opening are the parallel printer and games adapter connectors. In the illustration in Figure 2.34, the game port connector is located above the parallel port connector. On other systems, the locations of the various connectors may vary. The last connector on the back panel is the VGA adapter's monitor connector. The 15-pin connector is the VGA-compatible RGB color output port.

Keyboards

The keyboard most widely used with desktop and tower units is a detachable, low-profile 101/102-key model depicted in Figure 2.35. These units are designed to provide the user with a high degree of mobility and functionality. The key tops are slightly concave to provide a comfortable feel to the typist. In addition, the key makes a noticeable tap when it bottoms out during a keystroke.

Figure 2.35

An alphanumeric keyboard.



The keys are divided into three logical groups according to their function. Along the top of the board are special function keys (F1–F12), which assume special functions for different software packages. On the right side of the board is a numeric keypad, which does double duty as cursor control keys, under the control of the numbers-lock (NUM LOCK) key. A separate set of cursor control keys can be found between the normal keyboard keys and the numeric keypad. When power is first applied to the keyboard, the keypad keys function as `cursor control` keys. If the NUM LOCK key is pressed once, the keys function as a numeric keypad. If pressed again, the keys revert to their cursor control functions.

The white keys in the center group normally function as a standard QWERTY typewriter keyboard. This is the standard typist's keyboard arrangement. The keyboard's gray keys provide control functions. In addition to the normal typewriter-like control keys (backspace, two shift keys, return key (also referred to as the carriage return or enter key) and a TAB key (arrows pointing to both sides), the keyboard has several computer-related control keys. These include two special shift keys (CTRL and ALT), three special shift-lock keys (CAPS LOCK, NUM LOCK, and SCROLL LOCK), and an ESC key. When any of the three shift-lock keys is engaged, its corresponding LED (in the upper-right corner of the board) lights up to show that the function is engaged.

The original PC and XT employed an 84-key version that did not include the separate numeric keypad keys. Laptops, notebooks, and sub-notebooks use a compact 83-key version with many dual-purpose keys. As with the full-size keyboards, these units feature the capability to redefine every key on the board through software. This is an outstanding advantage of the PC keyboard in that it enables the keyboard to assume any function the programmer desires.

Video Displays

Desktop and tower units normally use a color `cathode-ray tube` (CRT) display monitor, similar to the one shown in Figure 2.36, as standard video output equipment. The PC and PC/XT and PC-AT often used `monochrome` (single color) monitors. They could also use color monitors by simply adding a color video adapter card and monitor. The color CRT monitor is sometimes referred to as an RGB monitor, because the three primary colors that make a color CRT are red, green, and blue.

Figure 2.36

The CRT display monitor.

Modern units normally use high-resolution color monitors for output purposes. The resolution of a monitor refers to the quality of the displayed image. High-resolution monitors produce images with greater detail than lower-resolution monitors. Working with a low-resolution monitor for extended periods of time leads to chronic eye, neck, and back strain, and accompanying headaches. Reflected room light has also been shown to increase display-related fatigue.

The monitor can either be plugged into a commercial, three-prong power receptacle, or into the special receptacle provided through the power supply at the rear of the unit. This option depends on the type of power cable provided by the manufacturer. A special adapter cable is available to match a standard 120VAC plug to this power supply receptacle. The `signal cable` (connected to the video adapter card) enables the monitor to be positioned away from the system unit if desired. However, it is normal to place the display directly on top of the system unit.

The display's normal controls are brightness and contrast. These controls are located in different positions on the monitor depending on its manufacturer. A power ON/OFF switch is also on the monitor. Its location varies from model to model as well. If the monitor receives power through the system unit's power supply, the monitor's power switch can be set to ON and the monitor turns on and off along with the system.

Other Peripherals

A system may employ one or more `pointing devices` as input sources. Most of these devices connect to that system in the rear of the unit. Mice typically plug into a 9-pin or 25-pin male D-shell connector. Joysticks and game paddles are normally connected to the 15-pin female D-shell Game Port connector. Light pens usually require a special connector on the video display card.

Printers are the other most widely used peripheral. Serial interface versions normally plug into a 9-pin or 25-pin male D-shell connector. Most of the time, it is common to find a serial printer connected to the 25-pin connector that has been set up as the system's second serial port. The first serial port is usually set up with the 9-pin connector and handles the mouse connection. Parallel printers are connected to the 25-pin female connector at the rear of the system. These typical peripheral connections are depicted in Figure 2.37.

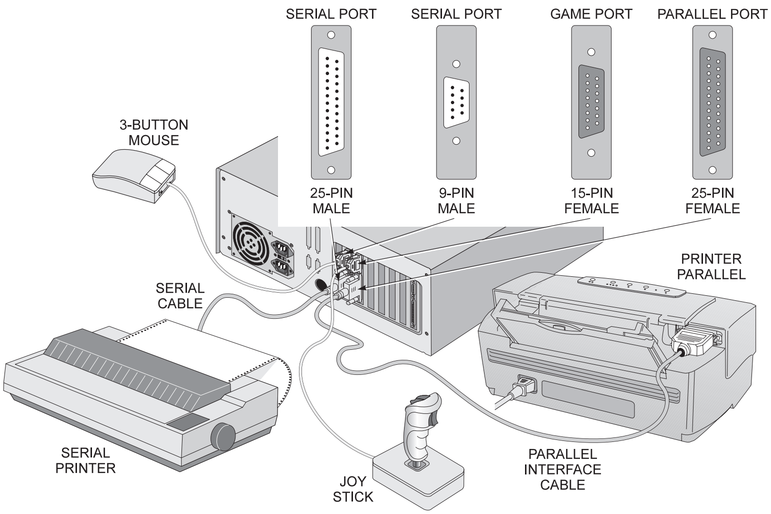


Figure 2.37

Typical peripheral connectors.

Summary

This chapter has covered the fundamental hardware structures and components associated with PC-compatible personal computer systems. You should be able to identify the major components of a typical personal computer and describe their functions. In addition, you should be able to differentiate between various styles of PCs (for example, desktop, tower, portable) and describe benefits and drawbacks associated with each style. Finally, you should be able to install, connect, and configure common computer hardware components to form a working system.

The next chapter begins to look at the other half of any computer system: its software. Chapter 3, “Operating Systems,” provides an in-depth examination of BIOS firmware and the bootup process, followed by an exploration of disk operating systems. This features a structured discussion of common MS-DOS commands.

Lab Exercise

A hands-on lab procedure corresponds to the theory materials presented in this chapter. Refer to Lab 1 on the CD-ROM and perform Procedure 1, "Orientation."

Review Questions

1. What type of IC is the brain of the PC system?
2. List four types of ICs commonly found on the system board.
3. List the four sub-units typically found inside the system unit.
4. How can you avoid confusion between the DB-15M connectors for VGA and game port connections?
5. List three types of memory typically found on modern system boards.
6. Where would you normally expect to encounter a PCMCIA card?
7. What is the purpose of a back plane? Where is it found?
8. List the devices normally found outside the system unit.
9. Name one weakness of tower cases.
10. Which government agency would be concerned with the characteristics of the computer's case?
11. Where is the system's BIOS program located?
12. How many floppy drives can a typical FDC controller handle? How are these drives identified to the system?
13. What type of port is indicated by the presence of a 9-pin male connector on the back panel of the computer?
14. Describe the major maintenance problem associated with notebook computers.

15. When purchasing a video card to install in a particular unit, what consideration must be taken into account?
16. Which 32-bit bus can accept cards from PC and ISA buses?
17. What type of ASIC device is normally found on a video adapter card and what are its functions?
18. When connecting a power supply to a system board, what precaution should be taken?
19. What does EFI stand for and why is it associated with a microcomputer system?
20. Where is the MI/O function normally found in a Pentium system?
21. What do the terms SIMM and SIPP stand for and what kind of devices are they?
22. What two expansion cards were considered as standard equipment in pre-Pentium PCs?
23. Describe the two input devices that are commonly included in PCs.
24. List the I/O functions associated with a common MI/O adapter.
25. How is upgrading a system with a Flash ROM BIOS different from upgrading a system with a standard ROM BIOS?

Review Answers

1. The microprocessor. For more information, see the section "System Boards."
2. The microprocessor, RAM, ROM, and support ICs. For more information, see the section "System Boards."

3. The components inside the system unit can be divided into four distinct sub-units: a switching power supply, the disk drives, the system board, and the options adapter cards. For more information, see the section "The PC System."

4. The pins of this game port's 15-pin D-Shell are arranged differently from those in the 15-pin VGA connector. In the VGA connector, the pins are arranged in three rows; the pins in the game port are arranged in two rows. For more information, see the section "Multi-I/O Adapter Cards."

5. ROM, RAM, and cache memory. For more information, see the section "Primary Memory."

6. In a notebook computer. For more information, see the section "Portables."

7. The back plane is an extension of the system's expansion slots. It is designed to enable I/O cards to be mounted horizontally in the machine so that the system unit height can be minimized. For more information, see the section "Desktops."

8. The keyboard, the mouse, and the monitor. For more information, see the section "Peripherals."

9. Air flow through tower cases is generally not good. Additional fans are almost always installed in tower cases to overcome this drawback. For more information, see the section "Towers."

10. The FCC is responsible for electrical interference that might be generated by computer equipment in a nonapproved case. For more information, see the section "Desktops."

11. Typically in the ROM BIOS IC or ICs located on the system board. For more information, see the section "Primary Memory."

12. The normal FDC controller can control two floppy disk drives that the system sees as Drives A: and B:. For more information, see the section "Multi-I/O Adapter Cards."

13. A serial communications port. For more information, see the sections “Multi-I/O Adapter Cards” and “Other Peripherals.”

14. Nonstandard printed circuit boards, proprietary battery case designs, and difficult internal structure, all creating difficult access to many parts of the system. For more information, see the section “Portables.”

15. That it fits securely in the system board’s expansion slot connector, that it covers the slot opening in the back panel, and that it provides a standard VGA connector for the type of monitor it serves. For more information, see the section “Adapter Cards.”

16. The EISA bus is designed so that it can accept PC-bus and ISA expansion cards. For more information, see the section “32-Bit Slots.”

17. The integrated video controller IC. It is a microprocessor-like chip that oversees the operation of the entire adapter. For more information, see the section “Video Adapter Cards.”

18. That the black wires from the P8 and P9 connectors are side by side. For more information, see the section “Power Supplies.”

19. Electromagnetic field interference. This is unwanted radio interference that can escape from an improperly shielded case. For more information, see the section “Desktops.”

20. The MI/O function is typically an integral part of the Pentium System Board. A single ASIC device on the board handles the complete PC-compatible MI/O function. For more information, see the section “Adapter Cards.”

21. SIMM stands for single in-line memory module; SIPP stands for single in-line pin package. Both devices are types of RAM memory modules. For more information, see the section “Primary Memory.”

22. The Multi I/O card and video adapter card. For more information, see the section “Adapter Cards.”

23. The keyboard and mouse. For more information, see the section “Peripherals.”

24. A typical MI/O adapter provides a floppy-disk drive controller (FDC), a hard-disk drive controller/interface (HDC), a game port, a parallel printer port, and two serial ports. For more information, see the section “Multi-I/O Adapter Cards.”

25. The Flash ROM feature enables new information to be transferred into the ROM chip from the system. With a standard ROM BIOS, either the IC must be replaced or the entire system board must be replaced. For more information, see the section “Primary Memory.”

Chapter 3

Operating Systems

Upon completion of this chapter and its related LabExercises, you should be able to:

Objectives

- Describe the series of events that occur when power is applied to the system.
- Define multi-user, multi-tasking, and multi-processor operations.
- Explain the function of the system BIOS.
- Discuss the sequence of the events in the POST test.
- Discuss naming conventions as they apply to various types of files.
- List the events that occur during the bootup process.
- Describe the function and purpose of DOS.
- Describe methods of bypassing and correcting inoperable DOS startup sequences.
- Install and configure operating systems to the basic operational level.
- Install and configure application software packages.
- Configure the system through CMOS Setup procedures.
- Create, delete and navigate through directories.
- Find, copy, rename, delete, and move, files.

continues

- Manipulate file attributes in a DOS system.
- Use the AUTOEXEC.BAT and CONFIG.SYS files to optimize system performance.
- Load driver software for devices added to the system.
- Edit the AUTOEXEC.BAT and CONFIG.SYS files for troubleshooting purposes.

Introduction

The general responsibilities of an operating system were described in Chapter 1, "Microcomputer Fundamentals." In this chapter, you will investigate operating systems in greater depth. The first half of the chapter deals with the foundation of the operating system—the BIOS. This topic is discussed in four sections: power-on self-tests and system initialization, booting up to the operating system, system configuration, and BIOS functions.

The second half of the chapter deals with `Disk Operating Systems` (DOS). In this section, the structure of DOS systems is explored along with typical DOS disk organization. The commands and utilities available through the DOS command line are also presented.

Operating Systems

Objective

Literally thousands of different operating systems are in use with microcomputers. The complexity of each operating system typically depends on the complexity of the application the microcomputer is designed to fill. The operating system for a fuel mixture controller in an automobile is relatively simple; an operating system for a multi-user computer system that controls many terminals is relatively complex.

The complete operating system for the fuel controller can be stored in a single, small ROM device. It takes control of the unit as soon as power is applied, resets the system, and tests it. During normal operation, the operating system monitors the sensor inputs for accelerator setting, humidity, and so forth, and adjusts the air/fuel mixing valves according to predetermined values stored in ROM. The fuel mixture controller is depicted in Figure 3.1.

In the large, multiple-user system, the operating system is likely to be stored on disk and have sections loaded into RAM when needed. As illustrated in Figure 3.2, this type of operating system must control several pieces of hardware, manage files created and used by various users, provide security for each user's information, and manage communications between different stations. The operating system is also responsible for presenting each station with a user interface that can accept commands and data from the user. This interface can be a command line interpreter or a graphical user interface (GUI).

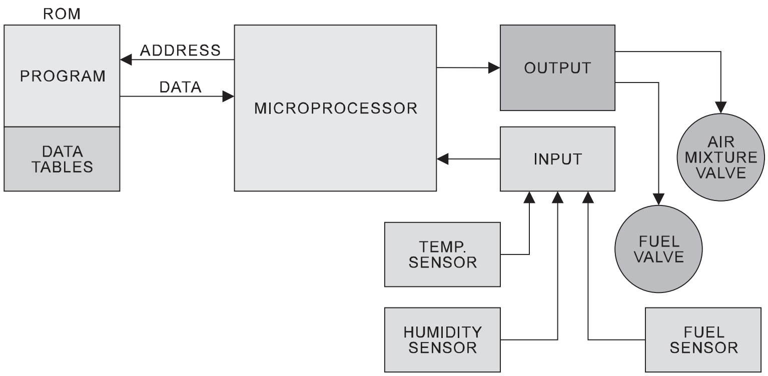


Figure 3.1

A simple air/fuel mixture controller.

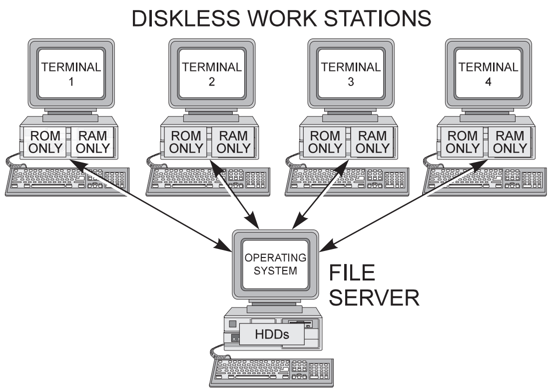


Figure 3.2

A multi-user system.

Complex operating systems typically contain several millions of lines of computer instruction. Due to this complexity, large operating systems are typically written in modules that handle the various responsibilities assigned to the system. The operating system for the fuel mixture controller is most likely a single module. However, the operating system for the multiple-user system probably consists of a core module, called the `kernel`, a task manager, a scheduler, a local file manager, and a host of other manager modules.

There are two basic types of operating systems:

- Single-process systems
- Multiple-process systems

In a single-process system, the operating system works with a single task only. These operating systems can operate in `batch mode` or `interactive mode`. In batch mode, the operating system runs one program until it is finished. In interactive mode, the operation of the program can be modified by input from external sources. The simple program presented with the $1.98 Computer first introduced in Chapter 1, "Microcomputer Fundamentals," is an example of a batch mode operating system. If a mechanism is added to the $1.98 so that program jumps can be caused by data from an external entry during the execution of the program, it then becomes an interactive system.

In multiple-process systems, the operating system is designed so that it can appear to work on several `tasks` simultaneously. A task is a portion of a program under execution. Computer programs are made up of several tasks that may work alone or as a unit. Tasks, in turn, can be made up of several `threads` that can be worked on separately. A thread is a section of programming that can be time-sliced by the operating system to run at the same time that other threads are being executed.

The multiple-process system breaks the tasks associated with a process into its various threads for execution. Typically, one thread might handle video output, another mouse input, and another output from the printer.

Objective

Multiple-process operations can be organized in three different ways:

- Multi-user
- Multi-tasking
- Multi-processor

These three types of operating systems are described in Figure 3.3.

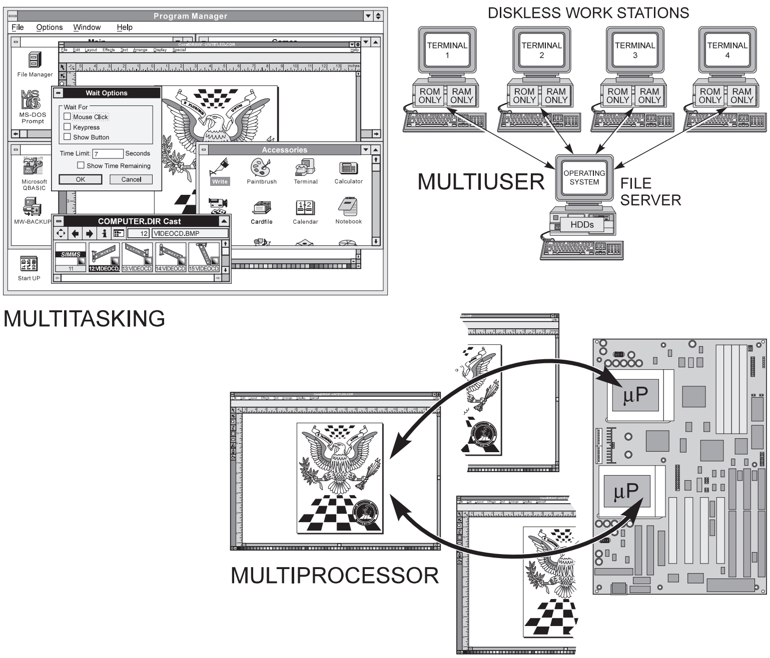


Figure 3.3

Multiple-process operating systems.

Multi-user and multi-tasking operations give the appearance of simultaneous operation by switching between different tasks in a predetermined order. The multi-user system switches between different users at multiple locations, while multi-tasking systems switch between different applications at a single location. In both cases, the information concerning the first task must be stored and information about the new task loaded each time a task switch occurs. The operating system's scheduler module is responsible for overseeing the switching function.

In a multi-processor operating system, tasks are divided between multiple microprocessors. This type of operation is referred to as parallel processing.

Although simple microcomputers store the entire operating system in ROM, most microcomputers use a bootstrapping process to load the operating system into RAM. Bootstrapping describes an arrangement in which the operating system is loaded into memory by a smaller program called the `bootstrap loader`. The operating system can be loaded from a ROM chip, a floppy disk, a hard-disk drive, or from another computer. The term `bootstrap` refers to the system pulling itself up by its own bootstraps, because in loading the more powerful operating system files from the disk, it has increased its on-board intelligence considerably. In personal computers, the bootstrap operation is one of the functions of the ROM BIOS.

Basic Input/Output Systems

Objective

PC system boards use one or two IC chips to hold the system's BIOS firmware. The system's memory map reserves memory locations from E0000h to FFFFFh for the system board BIOS routines. These chips contain the programs that handle startup of the system, the changeover to disk-based operations, video and printer output functions, and a `power-on self-test` (POST).

Post Tests and Initialization

Objective

The POST test is actually a series of tests that are performed each time the system is turned on. The different tests check the operation of the microprocessor, the keyboard, the video display, the floppy- and hard-disk drive units, as well as both the RAM and ROM memory units.

When the system board is reset, or when power is removed from it, the system begins generating clock pulses when power is restored. This action applies a RESET pulse to the microprocessor, causing it to clear most of its registers to 0. However, it sets the Instruction Pointer register to 0FFF0h and the CS register to F0000h. The first instruction is taken from location FFFF0h. Notice that this address is located in the ROM BIOS program. This is not coincidental.

When a cold boot is performed on the system, the microprocessor must begin taking instructions from this ROM location to initialize the system for operation.

Initial POST Checks

The first instruction that the microprocessor executes causes it to jump to the POST tests where it performs standard tests such as, the `ROM BIOS checksum test` (that verifies that the BIOS program is accurate), the system's various `DRAM tests` (that verify the bits of the memory), as well as testing the system's CMOS RAM (to make certain that its contents have not changed due to a battery failure). During the memory tests, the POST displays a running memory count to show that it is testing and verifying the individual memory locations.

Sequentially, the system's interrupts are disabled, the bits of the microprocessor's flag register are set, and a Read/Write test is performed on each of its internal registers. The test program simply Writes a predetermined bit pattern into each register and then Reads it back to verify the register's operation. After verifying the operation of the microprocessor's registers, the BIOS program begins testing and initializing the rest of the system. It moves forward by inspecting the ROM BIOS chip itself. It does this by performing a checksum test of certain locations on the chip, and comparing the answer with a known value stored in another location.

A checksum test involves adding the values stored in the key locations together. The result is a rounded-off sum of the values. When the checksum test is performed, the sum of the locations is recalculated and compared to the stored value. If they match, no error is assumed to have occurred. If not, an error condition exists and an error message or beep code is produced.

At this point, the program checks to see whether the system is being started from an off condition, or being reset from some other state. When the system is started from an off condition, a `cold boot` is being performed. However, simultaneously pressing the CTRL, ALT and DEL keys while the system is in operation generates a reset signal in the system and causes it to perform a shortened bootup routine. This operation is referred to as a `warm boot`, and enables the system to be shut down and restarted without turning it off. This function also enables the computer's operation to be switched to another operating system.

If power was applied to the system prior to the occurrence of the RESET signal, some of the POST's memory tests are skipped.

If a cold boot is indicated, the program tests the first 16KB of RAM memory by writing 5 different bit patterns into the memory, and reading them back, to establish the validity of each location. The BIOS startup steps are illustrated in Figure 3.4.

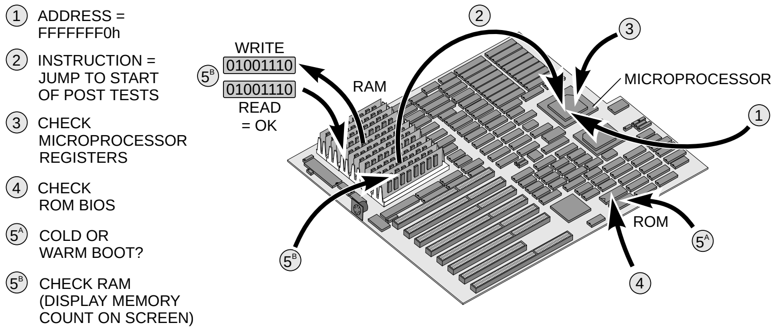


Figure 3.4
The startup sequence.

System Initialization

If the first 16KB of RAM successfully passes all five of the bit-pattern tests, the BIOS routine initializes the system's intelligent devices. During this part of the program, startup values stored in the ROM chip are moved into the system's programmable devices to make them functional. The BIOS initializes all the system's standard AT-compatible components, such as the interrupt, DMA, keyboard, and video controllers, along with its timer/counter circuits. The program checks the DMA controller by performing a R/W test on each of its internal registers, and then initializes them with startup values.

The program continues by setting up the system's Interrupt Controller. This includes moving the system's interrupt vectors into address locations 00000h through 003FFh. In addition, a R/W test is performed on each of the interrupt controller's internal registers. The routine then causes the controller to mask (disable) all its interrupt inputs, and tests each one to assure that no interrupts occur.

The programming of the interrupt controller is significant because most of the events in a PC-compatible system are interrupt driven. Its operation affects the operation of the computer in every phase from this point forward. Every peripheral or software routine that needs to get special services from the system makes use of the interrupt controller.

Following the initialization of the interrupt controller, the program checks the output of the system's Timer/Counter channels. It does this by counting pulses from the counters for a given period of time, to verify that the proper frequencies are being produced.

If the timer/counter frequencies are correct, the routine initializes and starts the video controller. The program obtains information about the type of display (monochrome, color, or both) being used with the system by reading configuration information from registers in the system's CMOS RAM. After this has been established, the program conducts R/W tests on the video adapter's RAM memory. If the video adapter passes all these tests, the program causes a cursor symbol to be displayed on the monitor. The steps of the initialization process are described in Figure 3.5.

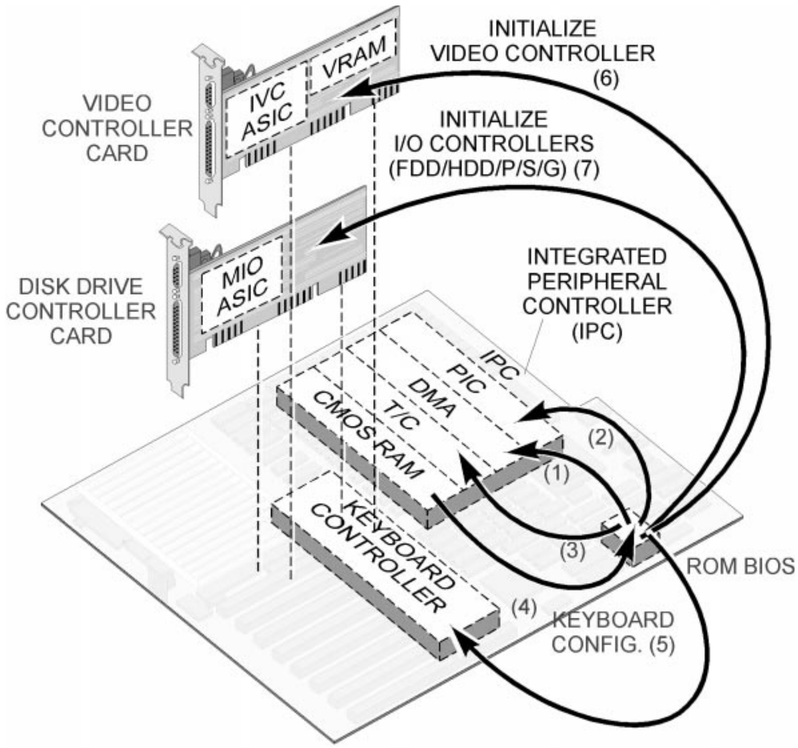
Additional POST Checks

After the display adapter has been checked, the BIOS routine resumes testing the system's on-board memory. First, R/W testing is performed on all the additional RAM on the system board (beyond the first 16KB). In addition, the BIOS executes the system's built-in setup program to configure its Day/Time setting, its hard- and floppy-disk drive types, and the amount of memory actually available to the system.

Following the final memory test, the remaining I/O devices and adapters are tested. The program begins by enabling the keyboard circuitry and checking for a scan code from the keyboard. No scan code indicates that no key has been depressed. The program then proceeds to test the system's parallel printer and RS-232C serial ports. In each case, the test consists of performing R/W tests on each of the port's registers, storing the addresses of functional ports (some ports may not be installed, or in use), and storing time limitations for each port's operation. The steps of the POST process are described in Figure 3.6.

Figure 3.5

System initialization.



BIOS Extensions

After the initialization and POST tests are completed, the BIOS checks the area of memory between C0000h and DFFFFh for `BIOS extension` programs. IBM system designers created this memory area so that new or non-standard BIOS routines could be added to the basic BIOS structure. These extended firmware routines match software commands from the system to the hardware they support. Therefore, the software running on the system does not have to be directly compatible with the hardware.

BIOS Extensions are created in 512-byte blocks that must begin at a 2KB marker (for example, C8000h, C8200h, C8400h, C8800h, and so forth), as illustrated in Figure 3.7. A single extension can occupy multiple blocks; however, it can start only at one of the markers. When the main BIOS encounters the special 2-byte extension code at one of the 2KB markers, it tests the block of code and then turns control over to the extension. Upon completion of the extension code, control is passed back to the main BIOS which then checks for an extension marker at the next 2KB marker.

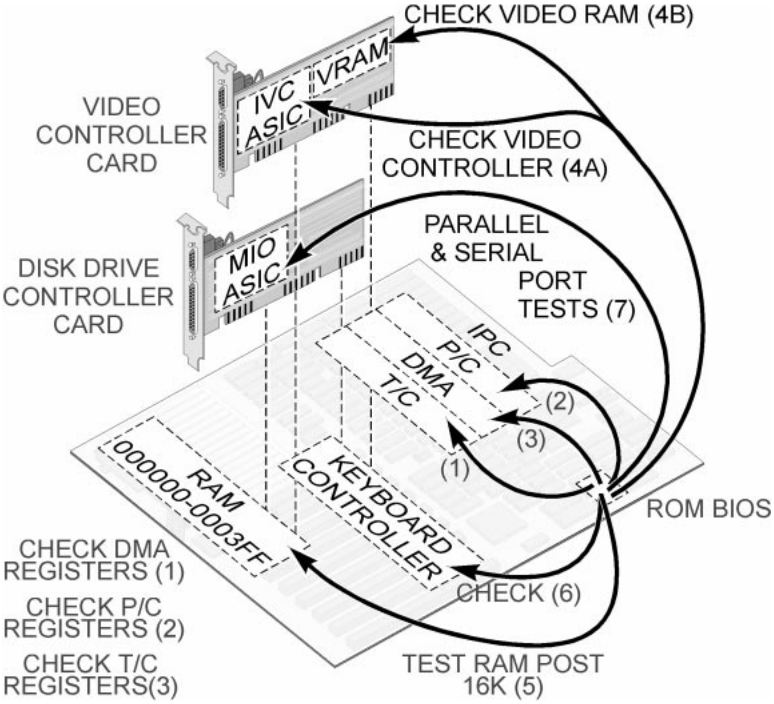


Figure 3.6

Completion of the POST test.

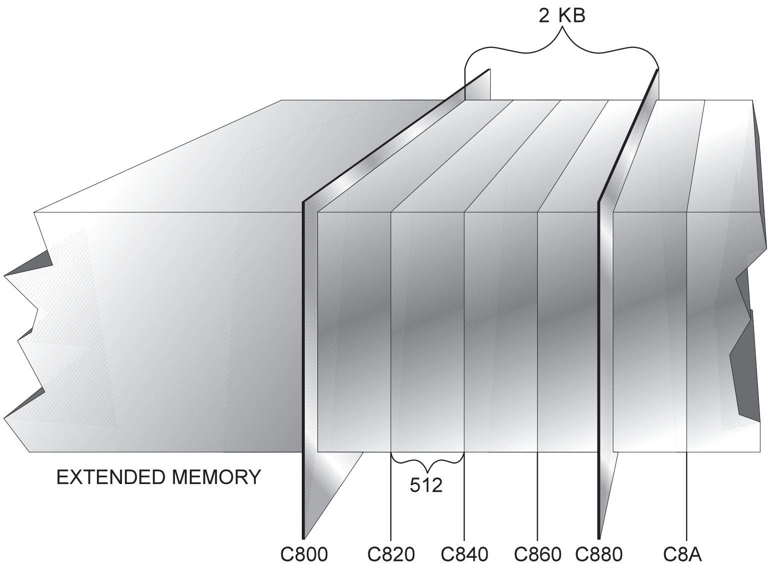
Although the extension addresses are memory addresses, the extension code may be located anywhere in the system. In particular, BIOS extensions are often located on expansion cards. The system simply accesses them through the expansion bus.

Advanced video cards contain Video BIOS code, either in a ROM IC, or built directly into the video controller ASIC. The IBM EGA and VGA standards allow for on-board ROM that uses addresses between C0000h and C7FFFh.

Likewise, different types of HDD controller cards contain a BIOS extension IC. The HDD controllers in old XT units had BIOS extensions that used the address space between C8000h and C9FFFh. Some current HDD controllers, such as ESDI and SCSI adapters (described in Chapter 8) reserve memory blocks between C8000h and CBFFFh.

Figure 3.7

BIOS extension blocks.



Another type of device that commonly uses the C000h-D000h blocks are network adapter cards. These cards enable the computer to be connected to other computers in the local area. The BIOS extension code on a network card may contain an `initial program load` (IPL) routine that causes the local computer to load up and operate from the operating system of a remote computer. Refer to the "Bootup" section of this chapter and the Networking information in Chapter 11 for more information about these BIOS extensions.

The system can accommodate as many extensions as fit mathematically within the allotted memory area. However, two extension programs cannot be located in the same range of addresses. With this in mind, peripheral manufacturers typically include some method of switching the starting addresses of their BIOS extensions so that they can be set to various markers.

CMOS Setup Routines

Objective

Just prior to completing the bootup process, older PCs and PC-XTs checked a set of configuration switches on the system board to determine which types of options were being used with the system. On newer systems, the configuration information is stored on the system board in a battery-powered storage area called the `CMOS RAM`. Newer BIOS enable the user to have access to this configuration information through the Setup utility.

While performing its normal tests and bootup functions, the BIOS program displays a header on the screen and shows the RAM memory count as it is being tested. Immediately following the RAM test count, the BIOS program places a prompt on the monitor screen to tell the user that the CMOS setup program can be accessed by pressing a special key, or a key combination. Typical keys and combinations include: the DEL key, the ESC key, the F2 function key, the CTRL and ESC keys, and the CTRL-ALT-ESC key combination.

Arguably, the most popular BIOS in the world are those from `American Megatrends, Inc.` (AMI). This BIOS uses the DEL key. Other BIOS programs may use different keys, or key combinations, for accessing their setup menus. If the DEL key is not depressed within a predetermined amount of time, the BIOS program continues with the bootup process. However, if the DEL key is pressed during this time, the bootup routine is put on hold and the program displays a "CMOS Setup Selection" screen, similar to the one depicted in Figure 3.8.

Every chipset variation has a specific BIOS designed for it. Therefore, functions specific to the design of system boards are using that chipset. Referring to the example in the figure, any one of three options can be selected from the screen: Return to the bootup process and continue normal operation, Select the Run CMOS Setup routine, or Run a built-in diagnostics program. This particular example is relatively simple. BIOS screens from other manufacturers or for other chipsets may have several options to consider. The example is also unusual in that it possesses a set of on-board diagnostic routines. These can be quite helpful when portions of the system are not functional, but they are not common in the industry.

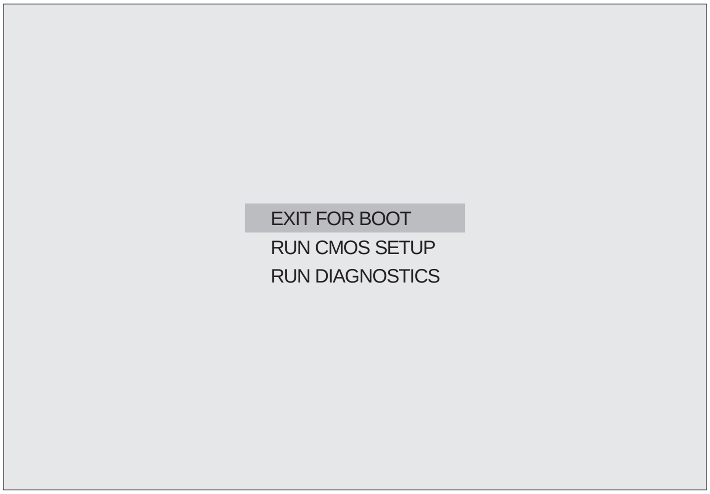


Figure 3.8

A CMOS Setup Selection screen.

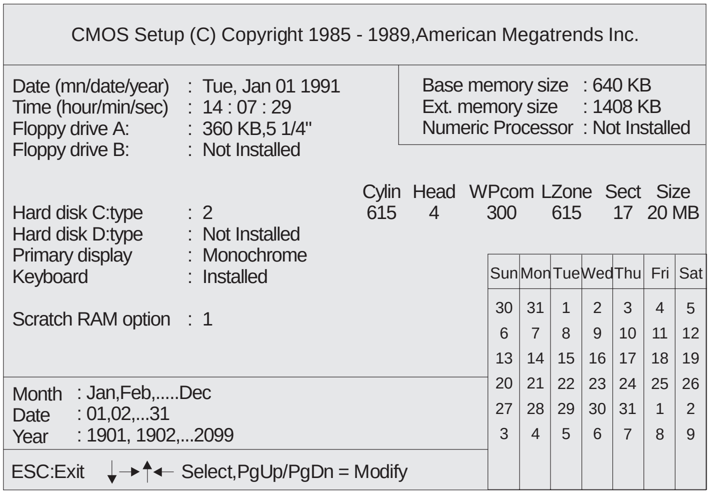
If you are setting the computer up for the first time or adding new options to the system, it is necessary to run the `CMOS Configuration Setup` program. The values input through the setup utility are stored in the system's CMOS Setup registers. These registers are examined each time the system is booted up to tell the computer which types of devices are installed.

The AMI Configuration Setup screen is shown in Figure 3.9. Through this screen, the user enters the desired configuration values into the CMOS registers. The cursor on the screen can be moved from item to item using the keyboard's cursor control keys.

When the cursor is positioned on top of a desired option, the PgUp and PgDn cursor keys can be used to change its value. When all of the proper options have been configured, pressing the ESC key causes the routine to exit the setup screen, update any changes made, and resume the bootup process.

Figure 3.9

The CMOS Configuration Setup screen.



CMOS Setup (C) Copyright 1985 - 1989,American Megatrends Inc.

Date (mn/date/year) : Tue, Jan 01 1991
Time (hour/min/sec) : 14 : 07 : 29
Floppy drive A: : 360 KB,5 1/4"
Floppy drive B: : Not Installed

Base memory size : 640 KB
Ext. memory size : 1408 KB
Numeric Processor : Not Installed

		Cylin	Head	WPcom	LZone	Sect	Size
Hard disk C:type	: 2	615	4	300	615	17	20 MB
Hard disk D:type	: Not Installed						
Primary display	: Monochrome						
Keyboard	: Installed						
Scratch RAM option	: 1						

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Month : Jan,Feb,.....Dec
Date : 01,02,...31
Year : 1901, 1902,...2099

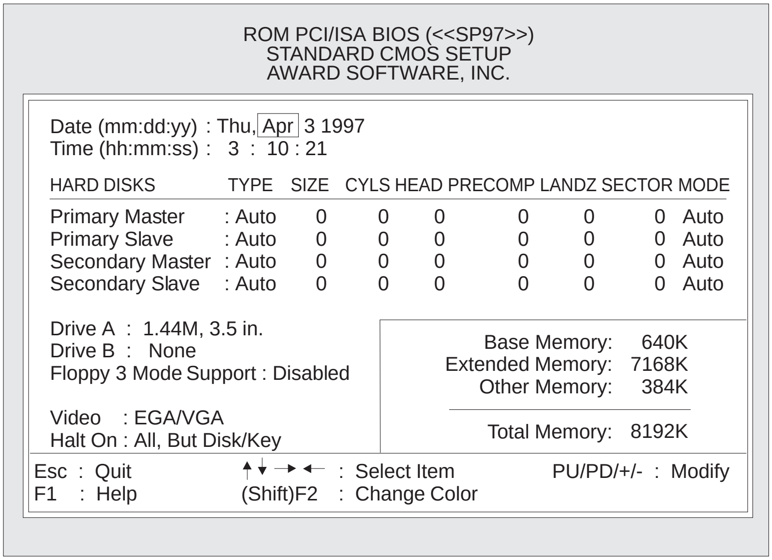
ESC:Exit ↓→↑← Select,PgUp/PgDn = Modify

Other BIOS Manufacturers

The Award BIOS from the Award Software company is another widely used BIOS. An Award BIOS Configuration selection menu screen is depicted in Figure 3.10. As the menu indicates, many user-configurable options are built into modern BIOS. Unlike the AMI BIOS, the Award firmware uses the + and – keys to manipulate the settings of menu items displayed on the screen.

Figure 3.10

The Award BIOS Configuration Setup screen.



ROM PCI/ISA BIOS (<<SP97>>)
STANDARD CMOS SETUP
AWARD SOFTWARE, INC.

Date (mm:dd:yy) : Thu, Apr 3 1997
Time (hh:mm:ss) : 3 : 10 : 21

HARD DISKS	TYPE	SIZE	CYLS	HEAD	PRECOMP	LANDZ	SECTOR	MODE
Primary Master	: Auto	0	0	0	0	0	0	Auto
Primary Slave	: Auto	0	0	0	0	0	0	Auto
Secondary Master	: Auto	0	0	0	0	0	0	Auto
Secondary Slave	: Auto	0	0	0	0	0	0	Auto

Drive A : 1.44M, 3.5 in.
Drive B : None
Floppy 3 Mode Support : Disabled

Video : EGA/VGA
Halt On : All, But Disk/Key

Base Memory: 640K
Extended Memory: 7168K
Other Memory: 384K

Total Memory: 8192K

Esc : Quit ↑↓→← : Select Item PU/PD/+/- : Modify
F1 : Help (Shift)F2 : Change Color

The standard CMOS setup screen is very similar to the AMI screen in Figure 3.9. In both examples, the BIOS first presents a screen of basic configuration information. As both figures show, this screen typically includes information about the time and date, microprocessor, system memory organization, floppy-disk drives, hard-disk drives, and video configurations.

A third major BIOS is produced by Phoenix Technologies, Ltd. Its main features are identical to the AMI and Award BIOS. The main screen covers time and date, hard- and floppy-disk drives, and system memory. The Phoenix BIOS uses the F2 function key to enter the Setup function's Main menu, depicted in Figure 3.11. Notice that the select keys for manipulating the Setup program are identified at the bottom of the display.

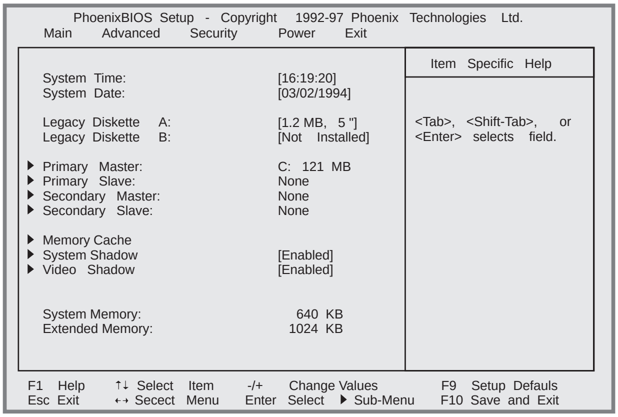


Figure 3.11

The Phoenix BIOS Configuration Setup screen.

In most CMOS displays, the total memory does not equal the summation of the base and extended memory. This is because the BIOS reserves 384KB for shadowing purposes.

The other area in this screen that typically requires some effort to set up is the HDD parameters section. All BIOS come with a list of hard drive types that they can support directly. However, they also provide a position for user definable drive settings. Historically, this has been referred to as the "Type 47" entry, but this entry may be located at any number in the list.

Advanced CMOS Setup

A second CMOS Configuration screen, referred to as the "BIOS Features Setup" screen, or "Advanced CMOS Setup" screen provides extended user control over the configuration of the system. A relatively simple Award Features screen is illustrated in Figure 3.12. In this example, several bootup options can be enabled, such as the boot drive sequence and Password enabling. The bootup sequence allows the system to bootup without checking all of the drives in order. This setting may need to be adjusted to include the A: floppy drive if it becomes impossible to boot the hard drive.

The password setting prevents users without the password from accessing the system. If the system has an unknown password, it will be necessary to clear the CMOS. Most system boards have a jumper block that can be shorted to reset the CMOS to its default settings. If this option is used, it will be necessary to reenter the original configuration information.

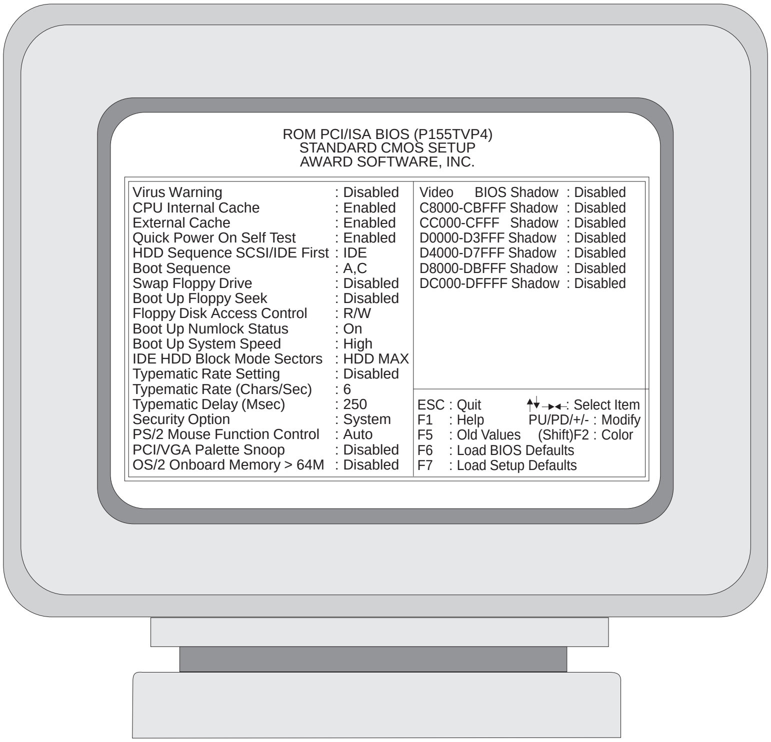


Figure 3.12

The Award BIOS features setup screen.

On Pentium-based system boards, the configuration jumpers and switches for enabling functions have been replaced by BIOS enabling settings. These settings usually include the disk drives, keyboard, and video options, as well as on-board serial and parallel ports. In addition, the user can turn certain sections of the system's RAM on or off for shadowing purposes, as well as establish parity or non-parity memory operation.

All of these enabling settings must be taken into account when troubleshooting the system's hardware. Incorrectly set BIOS enabling parameters will cause the corresponding hardware to fail. Therefore, check the enabling functions of the Advanced CMOS settings as a part of every hardware configuration troubleshooting procedure.

The complexity of modern system boards has created a huge number of configuration options for their BIOS. This is reflected in the complexity of their Advanced CMOS Configuration screens. Working in these screens, it is very easy to place the system in a condition where it is unable to respond. Because the problem is at the BIOS level, it is often difficult to get back into the CMOS to correct the problem. Therefore, system designers have included a couple of options to safeguard the system from this condition. In some BIOS, holding down the DEL key throughout the startup erases the CMOS contents and starts from scratch. There may also be jumpers placed on the system board that can be set to start the contents from a bare essentials setting. In either case, it is necessary to rebuild any advanced features in the CMOS configuration afterwards.

BIOS Entry and Exit

Even the Option selection pages for newer BIOS can be complex. A typical Options page is depicted in Figure 3.13. This screen serves as the main menu for entering and exiting the CMOS Setup, as well as for moving between its configuration pages.

BIOS designers have built two options into newer BIOS to help users avoid the complexity of the advanced CMOS configuration settings. These options are `Auto Configuration` and `Default Settings`.

All newer system boards have an auto-configuration mode that takes over most of the setup decisions. This option works well in the majority of applications. Its settings produce an efficient, basic level of operation for standard devices in the system. However, they do not optimize the performance of the system. To do that, it is necessary to turn off the auto configuration feature and insert desired parameters into the configuration table. The auto configuration function typically has two options: Auto Configure with Power-On Defaults and Auto Configure with BIOS Defaults.

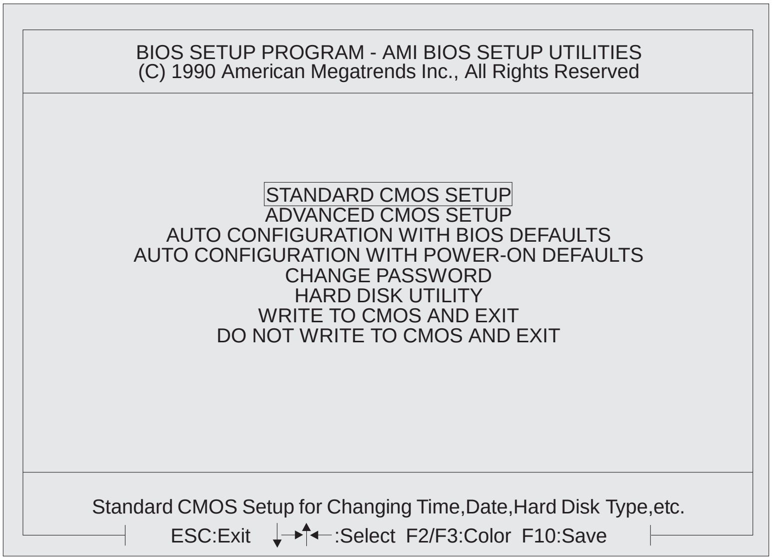


Figure 3.13

A complex entry menu.

Using Power-On defaults for auto-configuration loads the most conservative options possible into the system from the BIOS. This is the most effective method of detecting BIOS-related system problems. These settings replace any user entered configuration information in the CMOS Setup registers. Any Turbo speed mode is disabled, all memory caching is turned off, and all wait states are set to maximum. This allows the most basic part of the system to startup. If these default values fail to get the system to bootup, it is an indication of hardware problems such as, incorrect jumper settings or bad hardware components.

Using auto configuration with BIOS defaults provides a little more flexibility than the Power-On option. If you have entered an improper configuration setting and cannot determine which setting is causing the problem, this option is suggested. Like the Power-On option, this selection replaces the entered configuration settings with a new set of parameters from the BIOS. Choosing this option is likely to get you back into the CMOS Setup screen so that you can track down the problem. It is also the recommended starting point for optimizing the system's operation.

The many configuration options available in a modern BIOS requires the user to have a good deal of knowledge about the particular function that is being configured. Therefore, an extended discussion of the Advanced CMOS Setup options cannot be conducted at this point. However, such information is covered along with the system component it relates to as the book moves through various system components.

With older Award BIOS the CMOS Setup screen was accessed during bootup by pressing the ESC key. However, newer versions have adopted the same DEL-key strategy used with the AMI units. The exit routine is also different in that you can scroll between several exit options, or press the F10 key to save any changes, and exit the CMOS Setup. Older units required that the F5 key be pressed to confirm the exit selection. The newer units require a Y/N answer to exit.

Some BIOS may also offer a wide array of exit options. Typically, though, the options all involve writing the information away in CMOS and exiting, or not-writing the information to CMOS and exiting. One common mistake in working with CMOS configuration settings is not saving the new settings before exiting. When this happens, the new settings are not stored and the old settings are still in place when the system boots up.

BIOS Error Codes

If an error or setup mismatch is encountered, the BIOS issues an error code, either in message form on the display screen, or in beep-coded form through the system's speaker. Likewise, the Award BIOS produces display and beep-coded error messages when a bootup or configuration problem is encountered during the boot process.

In the case of Plug-and-Play systems, the BIOS must also communicate with the adapter cards located in the expansion slots to determine what their characteristics are. When the system is turned on, the PnP devices involved in the bootup process become active in their default configuration. Other logical devices, not required for bootup, start up in an inactive mode.

Before starting the bootup sequence, the PnP BIOS checks the devices installed in the expansion slots to see what types they are, how they are configured, and in which slots they are attached. It then assigns each adapter a `software handle` (name) and stores the names and configuration information in a RAM table. Next, the BIOS checks the adapter information against the system's basic configuration for resource conflicts. If no conflicts are detected, all the devices required for bootup are activated.

Objective

The devices not required for bootup may be configured and activated by the BIOS, or they may simply be configured and left in an inactive state. In either event, the operating system is left with the task of activating the remaining intelligent devices and resolving any resource conflicts that the BIOS detected and could not resolve. If the PnP option is not working for a particular device, or the operating system cannot resolve the remaining resource conflicts, then it will be necessary to use the Manufacturer's setup instructions to perform manual configurations.

Bootup

If the option to enter the Setup routine is bypassed, or if the routine has been exited, the BIOS begins the process of booting up to the operating system. A simple single operating system, single-disk bootup process is described in Figure 3.14. As you can see, it is a multiple-access operation that uses two different bootstrap routines to locate and load two different boot records.

The process starts when the BIOS begins looking through the system for a `master boot record`. This record can reside on drive A: or C:, or at any other location. The very first section on any logical DOS disk is called the `boot sector`. It contains information about how the disk is organized. It may also contain the small, optional master boot record that can access a larger, more powerful bootstrap loader program located elsewhere on the disk (normally in an area known as the root directory). In most systems, the master boot record is found at sector-0, head-0, and track-0 of the first logical hard drive. If the disk possesses a master boot record, it can boot up the hardware system to the operating system. The disk is then referred to as a `bootable disk`, or a `system disk`. If not, the disk is simply a `data disk` that can be used for storing information.

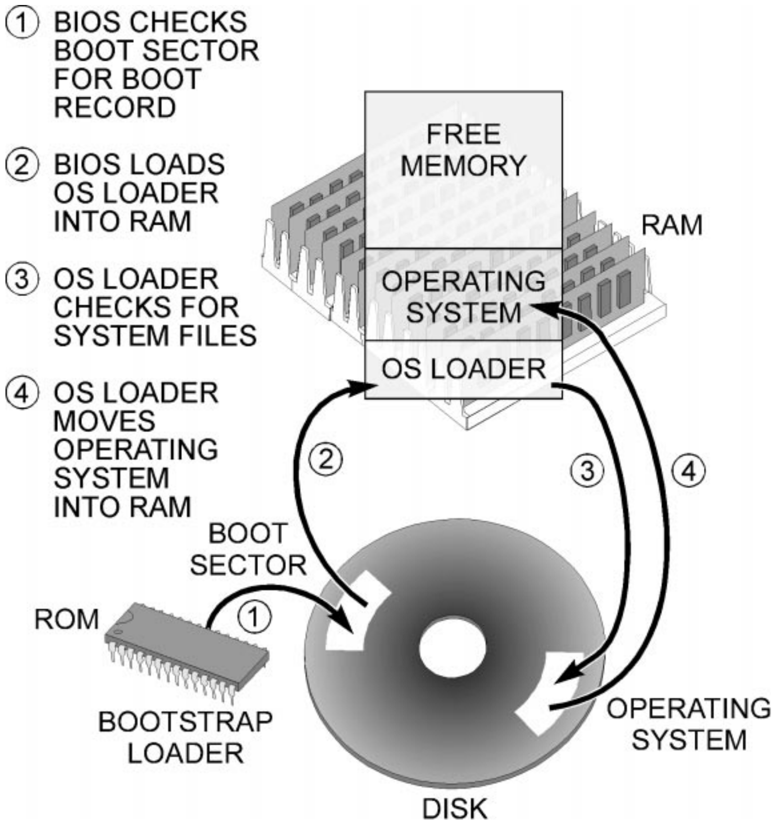


Figure 3.14

The bootstrap operation.

Traditionally, BIOS programs searches for the master boot record in floppy-disk drive A: first. If a bootable disk is in the floppy-disk drive, the BIOS executes the `primary bootstrap loader` routine to move the master boot record into RAM and then begin the process of loading the operating system. In the original IBM PC, the BIOS searched in the floppy-disk drive for the boot record. If it was not located there, the BIOS routine turned over control to a BASIC program located in the PC's ROM BIOS IC. In the PC-XT, the BIOS looked first in the floppy drive, or drives, and then in the hard-disk drive. If neither location contained the boot record, the system loaded up the ROM BASIC program. In clone systems, there was no ROM BIOS present to default to when no boot record was found. If the BIOS did not locate the boot record in the floppy or hard drive, it simply displayed a "Non-System Disk or Disk Error," or "ROM BASIC Interpreter Not Found" message on the screen.

In newer systems, the order in which the BIOS searches drives for the boot record is governed by information stored in the system's CMOS configuration RAM. The order can be set to check the floppy drive first and then the hard drive, or to check the hard drive first, or to check the hard drive only.

In a networked system, a bootstrap loader routine can also be located in the ROM extension of a network card as described earlier. When the system checks the BIOS extensions, the bootstrap routine redirects the bootup process to look for a boot record on the disk drive of another computer. Any boot record on the local drive is bypassed. Networking is covered in Chapter 11.

To accomplish the bootup, the BIOS enables the system's nonmaskable interrupts and causes a single, short tone to be produced by the speaker circuitry. The single beep indicates that the POST portion of the bootup has been successfully completed.

The next BIOS instruction executes an Interrupt19 Disk Drive service routine. This interrupt routine carries out the `primary bootstrap loader` program, which looks for the master boot record in the first section of the floppy and hard disks. When located, it moves the master boot record into system RAM to be executed.

The master boot record contains the `secondary bootstrap loader`, also called the `operating system loader`. This routine looks for an `operating system boot record`, typically located on the disk. When the routine finds the record, it loads the bigger boot record into RAM and begins executing it. This boot record brings special operating system files into memory so that they can control the operation of the system (for example, the operating system).

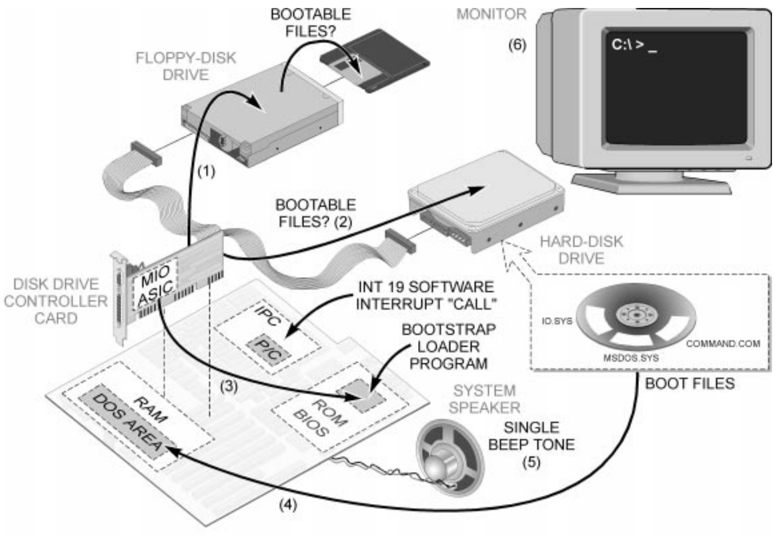
The operating system loader looks for a command processor file. The command processor can belong to any operating system, such as Microsoft MS-DOS, UNIX, IBM PC DOS, Novell Netware, and so forth. The default command processor for DOS is a system file called COMMAND.COM. This file interprets the input entered at the DOS prompt. When the bootstrap program finds the command processor, it moves it into system RAM along with the operating system support files.

In the case of Microsoft DOS, the special files in the OS boot record are the IO.SYS and MSDOS.SYS files. The BIOS recognizes these files by special extensions added to their names (.SYS and .COM).

In the original PC-DOS from IBM, the files were titled IBMBIO.SYS, IBMDOS.SYS, and COMMAND.COM. This step marks the end of the BIOS routine. The three system files must be found in the root directory (the starting point for any disk-based operations) in order for DOS to boot successfully. The total bootup process is described in Figure 3.15.

Figure 3.15

The bootup process.



If the system has performed a standard DOS bootup, without any modifications, it should print Date and Time prompts on the monitor screen, followed by the DOS command line “prompt” (A:\>> or C:\>>). The prompt indicates that DOS is operational and the currently active drive is the A: floppy drive, or the C:\ hard drive. Now the DOS software controls the movement of data, and overall operation of the system. DOS enables the basic bootup to be modified through two special utility files, called CONFIG.SYS and AUTOEXEC.BAT, discussed later in this chapter.

The operation of the system is now in the control of the operator, and whatever software is being used with the system. The system is waiting for the user to do something, such as enter commands and instructions, or run programs from the other two software categories. The user hasn't had anything to do with the operation of the system yet. This is why this type of software is referred to as system software.

BIOS Services

While the system is operating, the BIOS continues to perform several important functions. It contains routines that the operating system calls on to carry out basic services. These services include providing BIOS `interrupt CALLs` (software interrupt routines) for such operations as printer, video, and disk-drive accesses.

The ROM BIOS services are organized into groups identified by interrupt numbers. Each interrupt may cover several different services. When the microprocessor jumps to a particular interrupt, the software calling the interrupt must have already loaded the service number into the microprocessor to tell it which section of the interrupt handler to access.

The most notable BIOS interrupt calls include:

10h—Video Services (16)

13h—Hard and Floppy Drive Services (17 and 11)

14h—Serial Port Services (6)

16h—Keyboard Services (7)

17h—Parallel Printer Port Services (3)

18h—ROM BASIC (old systems)/Network Card Services (newer systems)

19h—Primary Bootstrap Loader

1Ah—Real Time Clock Services

The numbers in parenthesis refer to the number of different services available through the interrupt, For example, 10h—Video

Services (16) indicates that 16 different services are available through interrupt call 10.

This list represents just a few of the more notable BIOS interrupts. The most important thing for a technician to remember about BIOS interrupt CALLs is that they form the backbone of the system's operation. The BIOS and DOS are constantly handing control of the system back and forth between themselves as normal system functions are carried out. These BIOS interrupt CALLs are also responsible for most of the drawbacks of the PC system. That is why so much effort is exerted in software to work around them. Advanced operating systems implement newer methods of handling system functions just to avoid handing control over to the BIOS interrupts.

Some older PCs have trouble supporting newer hardware because the BIOS does not support the new item. To correct this situation, it is usually necessary to load a software driver program to support the device. Another possibility is to replace the BIOS with an improved version. This operation is not performed often because an upgraded BIOS must be compatible with the older chip set.

MS-DOS

MS-DOS is a disk operating system for IBM PC-compatible computers. It is easily the most popular operating system in the world. As with any other operating system, its function is to oversee operation of the system by providing support for executing programs, controlling I/O devices, handling errors, and providing the user interface. MS-DOS is a disk-based, single user, single task operating system.

The main portions of MS-DOS are the IO.SYS, MSDOS.SYS, and COMMAND.COM mentioned earlier. The IO.SYS and MSDOS.SYS files are special, hidden system files that do not show up in a normal directory listing. The IO.SYS file implements the MS-DOS default control programs (referred to as device drivers) for various hardware components. These include:

- Boot disk drive
- Console display and keyboard

- System's time-of-day clock
- Parallel and serial communications port

Conversely, the MSDOS.SYS file provides default support features for software applications. These features include:

- Memory management
- Character input and output
- Real time clock access
- File and record management
- Execution of other programs

The COMMAND.COM command interpreter contains the operating system's most frequently used commands. When a DOS command is entered at the DOS prompt, the COMMAND.COM program examines it to see whether it is an `internal` DOS command, or an `external` DOS command. Internal commands are understood directly by COMMAND.COM, while external commands are stored in a directory called DOS. If it is one of the internal commands, the COMMAND.COM file can execute it immediately. If not, COMMAND.COM looks in the \DOS directory for the command program.

Likewise, when DOS runs an application, COMMAND.COM finds the program, loads it into memory, and then gives it control of the system. When the program is shut down, it passes control back to the command interpreter.

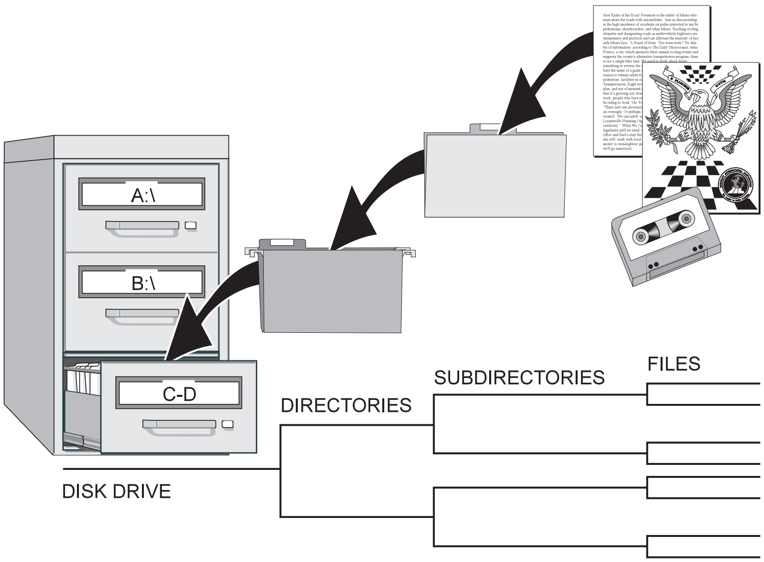
The remainder of the operating system is comprised of utility programs to carry out DOS operations such as formatting disks (Format), printing files (Print), and copying files (XCOPY).

MS-DOS Structure

It is important to consider that MS-DOS is a `disk` operating system. Therefore, you must understand how DOS organizes disks. The DOS organizational structure is typically described as being like a common office file cabinet, similar to the one depicted in Figure 3.16. Think of DOS as the filing cabinet structure. Our example has four drawers that can be opened. Think of these as `disk drives` labeled A, B, C/D, and E. Inside each drawer are hanging folders that can hold different types of items. Think of these as `directories`. The hanging folders may contain different types of items or other individual folders. Think of these individual folders as `sub-directories`. For organizational purposes, each hanging folder and each individual folder must have a unique label on it.

Figure 3.16

DOS organization.



Inside each hanging folder or individual folder are the items being stored. In a real filing cabinet, these items in the folders are usually documents of different types. However, pictures and tapes and other items related to the folder can also be stored in them.

Think of the items inside the folders as `files`. Disk-based systems manage data blocks by giving them filenames. Recall that a file is simply a block of logically related data, given a single name, and treated as a single unit. Like the contents of the folders, files can be programs, documents, drawings or other illustrations, sound files, and so on.

To find an item in the cabinet, you simply need to know which drawer, hanging folder and folder it is located in. This concept can be translated directly to the computer system. To locate a particular file, you simply need to know in which drive, directory, and sub-directory it is located. In MS-DOS the `path` to any file in the system can be written as a direction to the computer so that it knows where to find the file toward which it is being directed. This format for specifying a path is as follows:

```
C:\directory name\subdirectory name\file name
```

where the C: specifies the C disk drive. The directory, sub-directory, and file names would naturally be replaced by their real names. The `back slashes` (\) after each item indicate the presence of a directory or sub-directory. The first slash indicates a special directory, called the `root directory`, which is present on all DOS disks.

If the direction is to a file, the filename is always placed at the end of the path. MS-DOS allows for a basic file name of up to eight characters. It also allows for an extension of up to three characters. The extension is separated from the main portion of the file name by a period and is normally used to identify what type of file it is (that is the file name file1.`ltr` could be used to identify a letter created by a word processor).

Note

File name extensions are not actually required for most files. However, they become helpful in sorting between files in a congested system. You should be aware that the operating system reserves some three letter combinations, such as .COM and .SYS, for its own use. More information about file names and extensions is presented in the subsequent section concerning file-level DOS commands.

DOS Disk Structure

Objective

It is also important to understand how DOS sees disks. In the earlier section on booting up, it was mentioned that the first area on each DOS disk is the boot sector. Although all DOS disks have this sector, they do not all have the optional master boot record located in the sector. Only those disks created to be bootable disks have this record.

File Allocation Tables

The second section of a DOS disk is an area referred to as the `File`

`Allocation Table` (FAT). This area is a table of information about the condition of the disk. Basically, the system logs the use of the space on the disk in this table. In older versions of DOS, the amount of space dedicated to tracking the sectors on the disk was 16 bits. Therefore, only 65,536 sectors could be accounted for. This parameter limited the size of a DOS partition to 32MB (33,554,432 bytes).

To more effectively manage the space on the disk, newer versions of DOS divide the disk into groups of logically-related sectors, called *allocation units*, or *clusters*.

As described in Chapter 1, "Microcomputer Fundamentals," the sectors on a DOS disk hold 512 bytes each. On the other hand, files can be any length. Therefore, a single file may occupy several sectors on the disk. The DOS disk routine breaks the file into sector-sized chunks and stores it in a cluster of sectors. In this manner, DOS uses the cluster to track files rather than sectors. Because the file allocation table has to handle information for a cluster only, rather than for each sector, the number of files that can be tracked in a given length table is greatly increased.

The organization of a typical FAT is described in Table 3.1. The first two entries are reserved for DOS information. Each sector after that holds a value. Each value may represent one of three conditions. A value of 0 indicates that the cluster is empty and can be used for storage. Any number besides 0 or FFFh indicates that the cluster contains data and the number provides the location of the next cluster in a chain of clusters. Finally, a value of FFFh (or FFFFh in a 16-bit entry) indicates the end of a cluster chain.

Table 3.1

File Allocation Table structure.

Cluster Number	Contents
Cluster 0	Reserved for DOS
Cluster 1	Reserved for DOS
Cluster 2	3 (contains data go to cluster 3)
Cluster 3	4 (contains data go to cluster 4)
Cluster 4	7 (contains data go to cluster 7)

continues

Table 3.1 Continued

Cluster Number	Contents
Cluster 5	0 (free space)
Cluster 6	0 (free space)
Cluster 7	8 (contains data go to cluster 8)
Cluster 8	FFFh (end cluster chain)
Cluster 9	0 (free)
**	
Cluster x	0 (free)
Cluster y	0 (free)
Cluster z	0 (free)

On floppy disks, common cluster sizes are one or two sectors long. With hard disks, the cluster size may vary from 1 to 16 sectors in length. The FAT keeps track of which clusters are used and which ones are free. It contains a 12- or 16-byte entry for each cluster on the disk. The 12-byte entries are used with floppy disks and hard disks that are smaller than 17MB. The 16-byte entries are employed with hard-disk drives larger than 17MB. Obviously, the larger entries enable the FAT to manage more clusters.

In version-b of Windows 95, also referred to as `OSR2`, Microsoft supplied a 32-bit file allocation table system called `FAT32` to make efficient use of large hard drives (larger than 2GB). Under the previous FAT structure, large drives used large partitions, which, in turn, required large cluster sizes and wasted a lot of disk space.

The FAT32 format in OSR2 supports hard drives up to 2TB (terabytes) in size. FAT 32 uses 4KB cluster sizes for partitions up to 8GB in size.

In free clusters, a value of zero is recorded. In used clusters, the cluster number is stored. In cases where the file requires multiple clusters, the FAT entry for the first cluster holds the cluster number for the next cluster used to store the file. Each subsequent cluster entry has the number of the next cluster used by the file.

The final cluster entry contains an end-of-file marker code that tells the system that the end of the file has been reached.

These cluster `links` enable DOS to store and retrieve virtually any size file that fits on the disk. However, the loss of any link makes it impossible to retrieve the file and use it. If the FAT becomes corrupted, chained files can become `cross-linked` with each other, making them useless. For this reason, two complete copies of the FAT are stored consecutively on the disk under the DOS disk structure. The first copy is the normal working copy, while the second FAT is used as a backup measure in case the content of the first FAT becomes corrupted.

The Root Directory

The next section following the FAT tables is the disk's `Root Directory`. This is a special directory that is present on every DOS disk. It is the main directory of every logical disk, and serves as the starting point for organizing information on the disk. The location of every directory, subdirectory, and file on the disk is recorded in this table.

Each directory and subdirectory (including the root directory) can hold up to 512, 32-byte entries that describe each of the files in it. The first 8 bytes contain the file's name, followed by three bytes for its filename extension.

The next eleven bytes define the file's `attributes`. Attributes for DOS files include:

- Read Only
- System File
- Volume Label
- Subdirectory Entry
- Archive (backup) status

Two bytes are used to record the time the file was created or last modified. This is followed by two additional bytes that record the date the file was created or last modified.

The final four bytes are divided equally between the value for the `starting cluster number` and a `byte count number` for the file. Unlike the other information in the directory, the information associated with the last four bytes is not displayed when a directory listing is displayed on the screen.

Because each root directory entry is 32 bytes long, each disk sector can hold 16 entries. Consequently, the number of files or directories that can be listed in the root directory is dependent upon how many disk sectors are allocated to it. On a hard-disk drive, 32 sectors are normally set aside for the root directory. Therefore, the root directory for such a disk can accommodate up to 512 entries. A typical 3-1/2" 1.44MB floppy has 16 sectors reserved for the root directory and can hold up to 224 entries.

Figure 3.17 describes the organization of a DOS disk and illustrates the position of the Boot Sector, File Allocation Tables, and the Root Directory. The remainder of the disk is dedicated to data storage. On a floppy disk the logical structure normally has a group of files located under the root directory. Directory structures can be created on a floppy, but due to their relatively small capacity, this is not normally done. However, a hard drive is another matter. With hard drives, it is normal to organize the disk into directories and subdirectories as described earlier in this chapter.

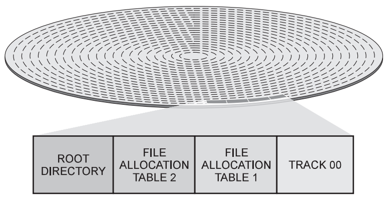


Figure 3.17

DOS disk organization.

Technically, every directory on a disk is a sub-directory of the root directory. All additional directories branch out from the root directory in a tree-like fashion. Therefore, a graphical representation of the disk drive's directory organization is called a `Directory Tree`. Figure 3.18 depicts the directory organization of a typical hard drive.

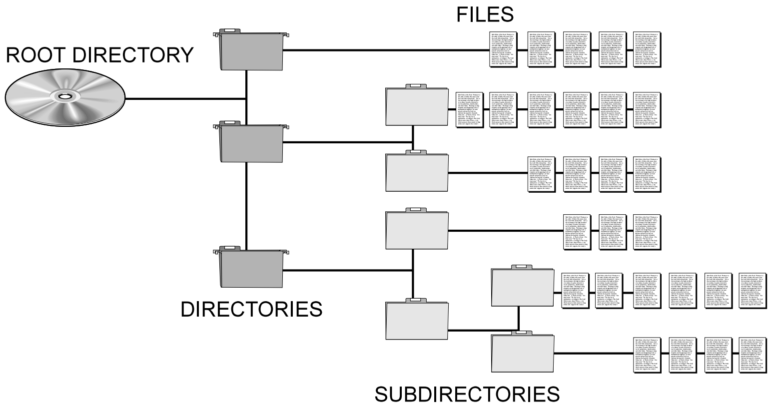


Figure 3.18

The DOS directory tree structure.

Under DOS, hard-disk drives can be divided into multiple `logical drives`. This operation is referred to as `partitioning` the drive. With earlier versions of DOS this became necessary as the capacity of hard drives exceeded the capability of the FAT to track all the possible sectors.

When a second logical drive is created on the hard disk, another boot sector, file allocation table and root directory is created. DOS sees this new structure on the hard drive as a completely new disk. Therefore it must have its own drive letter assigned to it.

In some applications, partitioning is popular because the system can be booted up to different operating systems. Because each partition contains its own boot sector, FAT, and Root Directory, each partition can be set up to hold and boot up a different operating system.

DOS Command Line

Objective

The operating system is responsible for providing the user interface. The main user interface for DOS is the command line. The command line is the space immediately following the DOS prompt on the screen. The MS-DOS prompt for using the C: hard-disk drive as the active directory is displayed in Figure 3.19.

Figure 3.19

The DOS prompt.

```
Mouse Version 8.00
1988 - 1993

Driver Installed : Mouse Systems Mode
Dynamic Resolution OFF
Mouse setup on COM1:

C:\MOUSE>
```

From the DOS prompt, all DOS functions can be entered and executed. Many programs are capable of being started from this prompt. These files can be discerned by their file name extensions. Files with .COM, .EXE, or .BAT extensions can be started directly from the prompt. The .COM and .EXE file extensions are reserved by DOS and can be generated by only programs that can correctly configure them. .BAT files are simply ASCII text files that have been generated using DOS functions. Because they contain DOS commands mixed with .COM and .EXE files, DOS can execute .BAT files from the command line.

Programs with other types of extensions must be `associated` with one of these three file types to be operated. The user can operate application software packages such as graphical user interfaces, word processors, business packages, data communications package, and user programming languages (such as QBASIC and DEBUG). As an example, the core component of a word processor could be a file called WORDPRO.EXE. Document files produced by word processors are normally given filename extensions of .DOC (for document) or .TXT (for text file). To view a document electronically, you first need to run the executable file and then use its features to load up, format, and display the document. Likewise, a BASIC file normally has an extension of .BAS assigned to it. To execute a file with this extension, it is necessary to run a BASIC interpreter, such as QBASIC.EXE, and then use it to load the .BAS file and then run it.

The user can also type DOS commands on the command line to perform DOS functions. These commands can be grouped into drive level commands, directory-level commands, and file-level commands. The format for using DOS commands is:

```
COMMAND (space) SOURCE location (space) DESTINATION location

COMMAND (space) location

COMMAND
```

The first example illustrates how DOS operations that involve a source and a final destination, such as moving a file from one place to another, are entered. The second example illustrates how single-location DOS operations, such as formatting a floppy disk in a particular disk drive, are specified. The final example applies to DOS commands that occur in a `default location`, such as obtaining a listing of the files on the current disk drive.

Many DOS commands can be modified by placing one or more software `switches` at the end of the basic command. A switch is added to the command by adding a space, a `fore-slash` (/), and a single letter:

```
COMMAND (space) option /switch
```

> Common DOS command switches include /P for page, /W for wide format, and /S for system. Different switches are used to modify different DOS commands. In each case, the DOS User's Guide should be consulted for switch definitions available with each command.

Drives and Disks

Objective

It is important to note that each disk drive in the system is identified by DOS with a single-letter name (such as A:), and that this name must be specified when giving the system commands, so that they are carried out using the proper drive. The format for specifying which drive is to perform a DOS operation calls for the presence of the drive's identifier letter in the command, followed by a colon (that is, A: or C:).

Figure 3.19 shows how the various disk drives are seen by a typical, stand-alone system. DOS assigns the letters A: and B: to the first and second floppy drives. Multiple hard-disk drive units can be installed in the system unit, along with the floppy drive(s). DOS recognizes a single hard-disk unit in the system as DRIVE C:. DOS utilities can also be used to divide a single, physical hard-disk drive into two or more volumes that the system recognizes as `Logical Drives` C:, D:, and so forth. This is known as `partitioning` the drive.

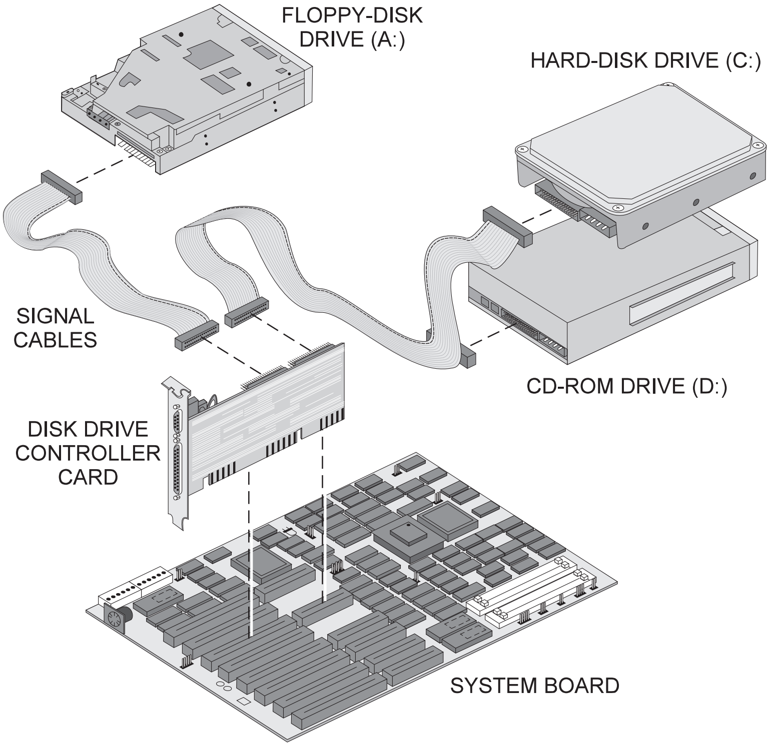
> Figure 3.20 shows a CD-ROM drive as Drive D: because this is becoming the most common PC configuration. In the case of networked systems, logical drive letters may be extended to define up to Z drives. These drives are actually the hard drives located in remote computers. The operating system in the local machine treats them as additional logical drives (for example, F, G, and so forth).

Conversely, a second hard-disk drive can be added to the system and set up as logical drive D:. It may also be partitioned into smaller logical drives that the system recognizes as drives E:, F:, and so on. Logical drives and disk partitioning are covered in Chapter 8.

Some DOS operations are simplified by enabling the system to choose the location for the command to be carried out through the use of `default settings` (special predetermined settings that are automatically used by the system when no specific directions are given to change the setting). These settings are remembered in DOS and used by the system when the operator does not specify a particular location for events to happen. The default setting in your system is the A: drive. In systems with two or more drives, it is imperative that the user specify exactly where the action called for is to occur.

Figure 3.20

The system's disk drives.



The following DOS commands pertain to drive-level operations. They must be typed at the DOS prompt, and carry out the instruction along with any drive modifiers given.

`DISKCOPY`: This command is used to make a duplicate of a disk. The DISKCOPY operation is normally used to make backup disks, and is usually followed by a DISKCOMP operation:

```
C:\>DISKCOPY A: B:
```

`DISKCOMP`: This command is used to compare the contents of two disks. It compares the data on the disks not only to see that they are alike, but also to see that the data is located in the same place on both disks. The DISKCOMP operation is normally used to check backup disks, and usually follows a DISKCOPY operation:

```
C:\>DISKCOMP A: B:
```

`FORMAT`: This command is used to prepare a new floppy disk for use. Actual data locations are marked off on the disk for the tracks and sectors, and bad sectors are marked. In addition, the directory is established on the disk. New disks must be formatted before they can be used.

`C:\>FORMAT B:` is used even in a single-drive system. The system issues prompts to insert the proper disks at the correct times. A self-booting disk can be created by using a /S modifier (for system files) at the end of the normal FORMAT command.

`C:\>FORMAT B:/S` causes three system files (boot files) to be copied onto the disk after it has been formatted. The new disk now boots up without a DOS disk.

`C:\>FORMAT A:/Q` causes the system to perform a quick format operation on the disk. This amounts to removing the FAT and root directory from the disk.

Directories

Objective

As mentioned earlier, in hard drive-based systems it is common to organize related programs and data into areas called `Directories`. This makes them easier to find and use, because modern hard drives are capable of holding large amounts of information. As described earlier, most directories can hold up to 512 directory or file name entries.

It would be difficult to work with directories if you could not know which one you were currently using. The DOS prompt can be set up to display which directory is being used. This particular directory is referred to as the `current` or `working directory` (for example, C:\DOS\forms indicates that you are working with programs located in a sub-directory of the DOS directory named forms). The first `back slash` (\) represents the root directory on the C hard drive. The presence of two dots (..) near the top of a directory listing acts to identify it as a sub-directory. These dots indicate the presence of a `parent directory` above the currently active subdirectory.

The following DOS commands are directory-based. The format for using them is identical to disk-related commands discussed earlier.

`DIR`: The `Directory` command gives a listing of the files on the disk that is in the drive indicated by the drive specifier.

`C:\>DIR` or `DIR B:` (If DIR is used without any drive specifier, the contents of the drive indicated by the prompt are displayed.) The command may also be used with modifiers to alter the way in which the directory is displayed.

`C:\>DIR/W` displays the entire directory at one time across the width of the display.

`C:\>DIR/P` displays the contents of the directory one page at a time. You must press a key to advance to the next display page.

`MKDIR (MD)`: creates a new directory in an indicated spot in the directory tree structure.

`C:\>MD C:\DOS\XXX` creates a new subdirectory named XXX in the path that includes the ROOT directory (C:\) and the DOS directory.

`CHDIR (CD)`: Changes the location of the active directory to a position specified with the command.

`C:\>CD C:\DOS` Changes the working directory from the C: root directory to the C:\DOS directory.

`RMDIR (RD)`: Remove directory erases the directory specified in the command. You cannot remove a directory until it is empty and you cannot remove the directory if it is currently active.

`C:\>RD C:\DOS\forms` removes the DOS sub-directory “forms,” provided it is empty.

`PROMPT:` Changes the appearance of the DOS prompt.

`C:\>PROMPT $P$G` causes the form of the prompt to change from simply C: to C:\> and causes the complete path from the main directory to the current directory to be displayed at the DOS prompt (for example, C:\DOS>).

`TREE:` Lists all the directory, and subdirectory, names on a specified disk.

`C:\>TREE C:` displays a graphical representation of the organization of the C hard drive.

`DELTREE:` Removes a selected directory and all the files and subdirectories below it.

`C:\>DELTREE C:\DOS\DRIVER\MOUSE` deletes the sub-directory "Mouse" and any sub-directories it may have.

Files and Filenames

Objective

Disk-based systems store and handle related pieces of information in groups called `files`. The system recognizes and keeps track of the different files in the system by their filenames. Therefore, each file in the system is required to have a filename that is different from that of any other file in the directory. If two files having the same name were present in the system at the same time, the computer would become confused and fail to operate properly, because it could not tell on which file it was supposed to work. Each time you create a new file of information, you are required to give it a unique filename by which DOS can identify it.

Under DOS, you must remember a few rules when you create new filenames. The filename consists of two parts: a `name` and an `extension`. The filename is a combination of alphanumeric characters and is between one and eight characters in length. The extension is an optional addition to the name that begins with a period, and is followed by between one and three characters. Extensions are not required on filenames, but they often prove useful in describing the contents of a file, or in identifying different versions of the same file. If a filename that already exists is used to store another file, the computer writes the information in the new file over that of the old file, assuming that they are both the same. Therefore, only the new file still exists. The information in the old file is lost.

Many software packages automatically generate filename extensions for files they create. The software does this so that other parts of the program, which may work with the same file, can identify the file's source location or its form.

In any event, you should remember the following items when assigning and using filenames:

- All files must have a filename.
- All filenames must be different than any other filename in the system, or on the disk presently in use.
- Filenames are up to 8 characters long with an optional 3-character extension (separated from the basic filename by a period).
- When using a filename in a command, you must also use its extension, if one exists.
- Some special characters are not allowed in filenames. These are the brackets, colon, semicolon, plus sign, equals sign, back slash, fore-slash, and comma ([,], :, ;, +, =, \, /, and ,).
- When telling DOS where to carry out a command, you must tell it on which disk drive the operation is to be performed. The drive must be specified by its letter name followed by a colon (for example, A:, B:, C:, and so forth).
- The complete and proper way to specify a file calls for the drive specifier, the filename, and the filename extension, in that order (for example, B:filename.ext).

The following DOS commands are used to manipulate specific files. The format for using them is identical to the disk-related commands discussed earlier. However, the command must include the filename and its extension at the end of the directory path. Depending on the operation, the complete path may be required, or a default to the currently active drive is assumed.

`COPY`: The file copy command copies a specified file from one place (disk or directory) to another.

```
C:\>COPY A:filename.ext B:filename.ext
```

`C:\>COPY A:filename.ext B:` is used if the file is to have the same name in its new location; the second filename specifier can be omitted.

In a single-drive system, it is necessary to switch disks in the middle of the operation. (Notice that the drive B specifier is used even though only drive A: is present.) Fortunately, the DOS produces a prompt message to tell you when to put the target disk in the drive. This is not required in a two-drive system and no prompt is given. The transfer can be specified in any direction desired, as in:

```
C:\>COPY B:filename.ext A:
```

The only thing to keep in mind in this situation is to place the source disk in drive B and the target disk in drive A: before entering the command.

`XCOPY:` This command copies all the files in a directory, along with any sub-directories and their files. This command is particularly useful in copying files and directories between disks with different formats (i.e., from a 1.2MB disk to a 1.44MB disk:

```
C:\>XCOPY A: B: /s
```

This command copies all the files and directories from the disk in drive A: (except hidden and system files) to the disk in drive B:. The /s switch instructs the XCOPY command to copy directories and sub-directories.

`DEL or ERASE`: When this command is typed in at the DOS prompt, it enables the user to remove unwanted files from the disk:

```
C:\>DEL filename.ext
```

```
C:\>ERASE B:filename.ext
```

A great deal of care should be taken when using this command. If a file is erased accidentally, it may not be retrievable.

`REN`: Enables the user to change the name or extension of a filename:

```
C:\>REN A:filename.ext newname.ext
```

Using this command does not change the contents of the file, only its name. The original filename (but not the file) is deleted. If you wish to retain the original file and filename, a copy command, using different filenames, can be used:

```
C:\>COPY A:filename.ext B:newname.ext
```

`TYPE`: Shows the contents of a designated file on the monitor screen.

`C:\>TYPE AUTOEXEC.BAT` displays the contents of the autoexec.bat file

`FC:` This file-compare command compares two files to see whether they are the same. This operation is normally performed after a file copy has been performed to ensure that the file was duplicated and located correctly:

```
C:\>FC A:filename.ext B:
```

If the filename was changed during the copy operation, the command would have to be typed as:

```
C:\>FC A:filename.ext B:newname.ext
```

`ATTRIB:` Changes file attributes such as read-only (+R or –R), archive (+A or –A), system (+S or –S), and hidden (+H or –H). The + and – signs are chosen to add or subtract the attribute from the file.

`C:\>ATTRIB +R C:\DOS\memos.doc` sets the file memos.doc as a read-only file.

Read-only attributes protect the file from accidentally being overwritten. Similarly, one of the main reasons for giving a file a Hidden attribute is to prevent it from accidentally being erased. The System attribute is reserved for use by the operating system and marks the file as a system file.

`SETVER`: This command sets the DOS version number that the system reports to an application. Programs designed for previous DOS versions may not operate correctly under newer versions unless the version has been set correctly:

```
C:\>SETVER C:
```

This entry causes all the files on the C: drive to be listed in the DOS version table. If the current DOS version is not known, typing VER at the DOS prompt displays it on the screen. These commands are particularly useful in networking operations where multiple computers are connected together to share information. In these applications, several versions of DOS may exist on different machines attached to the network.

DOS Shortcuts

Objective

DOS provides some command line shortcuts through the keyboard's function keys. Some of the most notable are the F1 and F3 function keys. The F1 key brings the preceding command back from the command line buffer, one character at a time. Likewise, the F3 key brings back the entire preceding command, through a single keystroke.

When using filenames in DOS command line operations, the filename appears at the end of the directory path in the source and destination locations. The * notation is called a `wild card` and enables operations to be performed with only partial source or destination information. Using the notation as *.* tells the software to perform the designated command on any file found on the disk using any filename and extension.

A question mark (?) can be used as a wild card, to represent a single character in a DOS name or extension. Multiple question marks can be used to represent multiple characters in a filename or extension.

Data from a DOS command can be modified to fit a prescribed output format, through the use of filter commands. The main filter commands are More, Find, and Sort. The filter command is preceded by a pipe symbol (|) on the command line, when output from another DOS command is to be modified. For example, to view the contents of a batch file that is longer than the screen display can present at one time, type **`TYPE C:\xxx.bat¦more`**. If the information to be modified is derived from another file, the less than (<<) symbol is used.

The Find command searches through files and commands for specified characters. Likewise, the Sort command presents files in alphabetical order.

DOS I/O Commands

Objective

The DOS mode command is used to configure the system's I/O devices. These devices include the parallel and serial ports as well as the monitor display and the keyboard.

DOS keeps track of its different parallel and serial ports by assigning them logical designations during the initialization phase of the system bootup. A parallel port is designated as an `LPT port` and can be assigned to the system as LPT1, LPT2, or LPT3. Likewise, serial ports are designated as `COM`, or `communications ports`. Any of the system's serial ports can be configured as COM1, COM2, COM3, or COM4. However, the serial ports cannot share the same COM port designation.

The format for using the mode command to configure the parallel printer port is as follows:

```
mode LPT1:n,m,P
```

where n is the number of characters per line across the page, m is the number of lines of print down the page, and the value of P sets up continuous retry on timeout errors (errors that occur when actions do not occur during a prescribed amount of time). The value of n can be set to 80 or 132 characters. Common values for m are 6 or 8 lines.

The mode command is also used to set up the serial ports. The format for the serial port is:

```
mode COMn:baud,parity,databits,stopbits,P
```

where n represents one of the four serial ports in the system. Baud is the transmission rate at which the port sends and receives data. Common values for this variable are 110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19,200. Only the first two digits of the speed rating are placed in the command (for example, 9600 = 96).

Parity describes the type of error checking used by the port (error checking and parity are discussed in Chapter 7, “Input/Output,” and Chapter 11, “Data Communications”). Parity can be set to E for even, O for odd, or N for none.

The data bit entry tells the receiver how many data bits to expect. The usual setting for data bits is 7; however, an 8-bit data word can also be selected. Likewise, different numbers of special stop bits can be used in serial communications to mark the end of a character or message. Typical stop bit values can be 1 or 2. The P value is used to indicate whether the port is being used with a serial printer or some other serial device. If a value is included for P, the system assumes that it is connected to a serial printer and performs continuous retries on timeout errors.

In addition to setting up the operation of the system’s I/O ports, the mode command can be used to alter the output format of the video display. The format for using the mode command to alter the output on the video display is:

```
mode n,m,T
```

where n is the number of columns and color selection for the display. Typical values for this variable are 40, 80, BW40, BW80, CO40, CO80, and mono. The 40 and 80 values indicate the numbers of characters on a text line. The BW40 and BW80 options also indicate the number of characters per line, but include a reference to the color graphics adapter with color turned off. Conversely, the CO40 and CO80 values indicate the color graphics adapter with color enabled. Mono indicates a monochrome display adapter.

The m variable can be set to r (for right shift), to l (for left shift). If the T value is present in the command, a test pattern is presented on the screen so that it can be aligned properly.

To use the mode command to set the display mode, a device statement must be included in the CONFIG.SYS file for the ANSI.SYS device driver.

Finally, the mode command can be used to shift data from one output port to another. As an example, it is possible to shift data intended for a serial port to the parallel port. This is a quick and useful troubleshooting tool when working with ports. If data intended for a suspect port can be successfully re-directed to another port, then a hardware problem with the first port is indicated. An example of re-directing data from one port to another is:

```
mode LPT1:=COM2
```

This example re-directs data intended for the first parallel port to the second parallel port.

The mode command can be used inside the AUTOEXEC.BAT file to automatically reconfigure the system at startup.

DOS Utilities

Objective

A subclass of system software, called `utilities`, can be used to perform some basic system operations. These programs enable the system to be optimized for operations in particular functions or with different options.

In the DOS operating system, two of these utilities, called the CONFIG.SYS and AUTOEXEC.BAT files, can be included in the DOS bootup process. As the system moves through the bootup procedure, the BIOS checks in the root directory of the boot disk for the presence of a file named CONFIG.SYS. Afterwards, it searches for the COMMAND.COM interpreter, and finally looks in the root directory for the AUTOEXEC.BAT file. In particular, the CONFIG.SYS and AUTOEXEC.BAT files play key roles in optimizing the system's memory and disk drive usage. This operation can be summarized as follows:

1. BIOS performs INT19 to search drives for master boot record.
2. Primary Bootstrap Loader moves master boot record into memory.
3. System executes Secondary Bootstrap Loader from master boot record.
4. Secondary Bootstrap loader moves IO.SYS and MSDOS.SYS into memory.
5. DOS checks for CONFIG.SYS file in root directory.
6. If CONFIG.SYS is found, DOS reconfigures system.
7. DOS loads COMMAND.COM.
8. COMMAND.COM checks for the AUTOEXEC.BAT file in the root directory.
9. If the AUTOEXEC.BAT file is found, COMMAND.COM carries out the commands found in the file.
10. If no AUTOEXEC.BAT file is found, COMMAND.COM displays the DOS Time and Date prompt as describe earlier.

DOS Memory

Objective

To understand how the CONFIG.SYS and AUTOEXEC.BAT files improve the performance of the system, you must understand how DOS views memory and why.

The original DOS version was constructed in two sections. The first 640KBs of memory was reserved for use by DOS and its programs. The remaining section was reserved for use by the BIOS and the system's peripherals (such as the video card, the hard drive controller card, and so forth). This arrangement utilized the entire 1MB addressing range of the 8088 microprocessor.

As more powerful microprocessors entered to market (80286 microprocessors can access up to 16MB of memory, and the 80386 and 80486 can handle up to 4GB of memory), DOS retained the limitations imposed on it by the original version to remain compatible with older machines and software.

Special add-on programs called `memory managers` have been created to enable DOS to access and use the additional memory available to more powerful microprocessors.

Basic Memory Organization

Every computer has a memory organization plan called a `memory map`. A simplified memory map, showing RAM, ROM, and I/O address allocations, is shown in Figure 3.21.

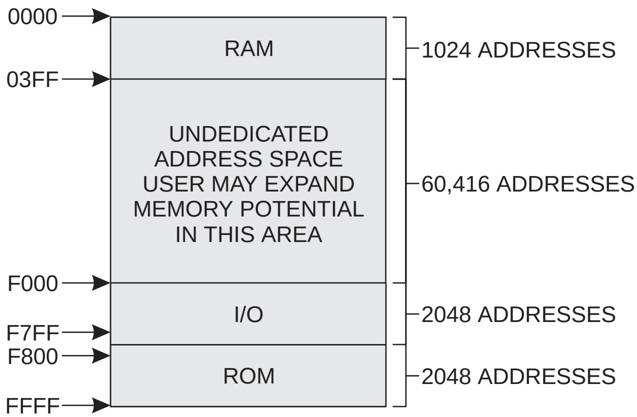


Figure 3.21

A computer memory map.

When the original PC was designed, there were certain decisions made in dividing up the 8088's 1MB of memory address space. The Intel microprocessors have a separate memory map for I/O addresses. These decisions were implemented by the original DOS program. By necessity, these decisions carried over into the address allocations of all DOS-based PC-compatible's.

Basically, DOS can recognize the following classifications of memory: Conventional Memory, Upper Memory Blocks, High Memory Area, Expanded Memory, Extended Memory, and Virtual Memory.

Conventional Memory

The conventional memory area is divided into two sections referred to as `Base Memory` and the `Upper Memory Area` (UMA). These sections are illustrated in Figure 3.22. Base memory occupies the first 640KBs of addresses while the remaining 384KBs is referred to as Upper Memory.

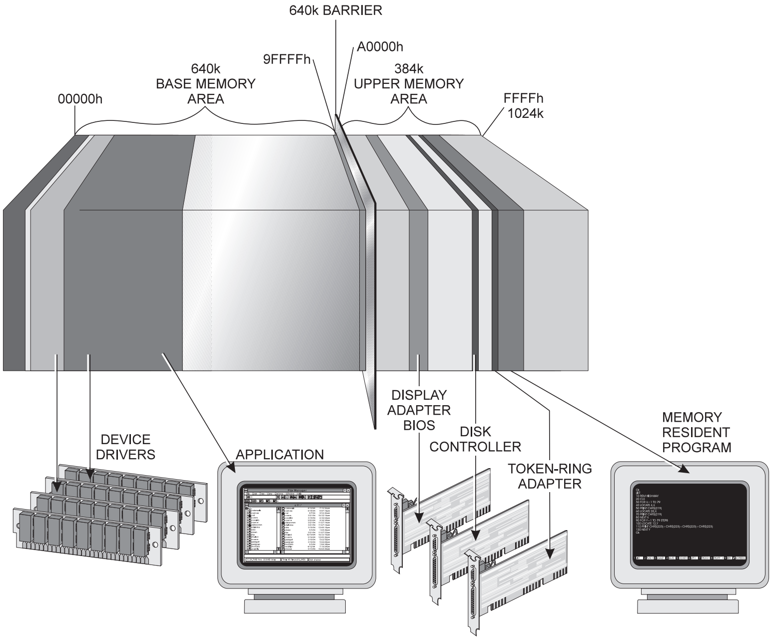


Figure 3.22

Conventional memory.

Base memory (locations 00000h through 9FFFFh) is the standard memory area for all PC-compatible systems. It traditionally holds DOS, interrupt vector tables, and relocated ROM BIOS tables. The remaining space in the base memory area is referred to as DOS Program Memory. Programs written to operate under PC- or MS-DOS use this area for program storage and execution.

The Upper Memory Area occupies the 384KB portion of the PC's address space from A0000h to FFFFFh. This space is segmented into 64KB `Upper Memory Block` regions, as illustrated in Figure 3.23. Although the addresses are allocated, no actual memory is here. The Upper Memory Area was originally dedicated to different forms of video display memory and ROM-based functions. However, many advanced systems reserve space in this area to incorporate a memory-usage scheme called `Shadow RAM` to improve the overall performance of the computer.

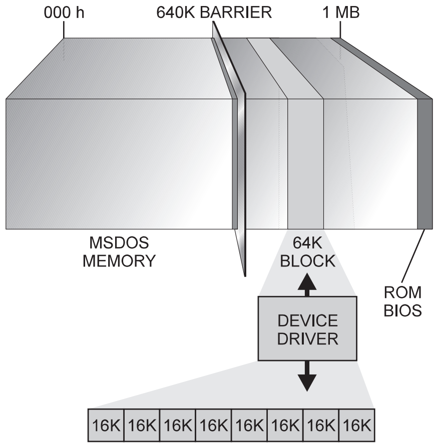


Figure 3.23

Upper memory blocks of the UMA.

With this feature, the contents of the ROM BIOS and/or Video BIOS are rewritten (shadowed) into Upper Memory Area. This scheme enables the system to operate faster when application software makes use of any of the BIOS' CALL routines. Rather than accessing an IC ROM device, which takes up to 4 Wait States to complete, BIOS calls are redirected by the shadow feature to the same information located in fast, 0 Wait State DRAM devices. Some benchmark tests have shown performance increases between 300 and 400% in systems where the shadow feature is used.

Extended Memory

With the advent of the 80286 microprocessor and its protected operating mode, it became possible to access physical memory locations beyond the 1MB limit of the 8088. Memory above this address is generally referred to as `Extended Memory`. With the 286 microprocessor, this adds up to an additional 15MB of RAM for a total of 16MB (24 bit address). Extended Memory is illustrated in Figure 3.24.

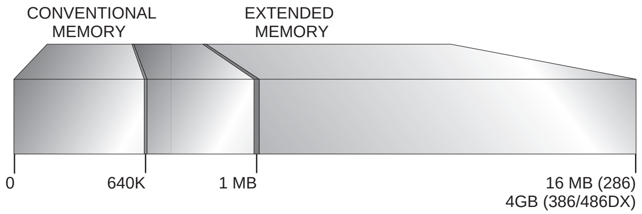


Figure 3.24

Extended memory.

Even though the 80286 could physically access this type of memory using its special addressing mode, it was impossible for application programs to access it at the time. This was due to the 640k DOS limit imposed by earlier architectures. Extended memory could range up to 4GB in 80386- and 80486-based computers (32 bit address). It was not that software couldn't access memory at these addresses, it was simply a matter of DOS not having the capability to enable it.

Applications programs can be written specifically to take advantage of these memory locations, but few are. Operating systems, such as Microsoft DOS versions beyond 4.0 and Windows versions beyond 3.0, as well as IBM's OS/2 operating system, can take full advantage of extended memory through the protected addressing modes of the more advanced microprocessors. This capability to manage higher memory enables the system to free up more of the base memory area for use by applications programs.

The DOS versions above 4.0 contain a memory management program called HIMEM.SYS that manages extended memory above the 1024k level. This utility operates under the Microsoft `Extended Memory Specification` (XMS). When the utility is loaded into memory, it shifts most of the operating system functions into an area known as the `High Memory Area` (HMA) of extended memory. The HMA takes up the first 64KB of addresses above the 1MB boundary and is a result of a quirk in the design of advanced Intel microprocessors.

The HIMEM function is activated by adding a line of instruction to the system's CONFIG.SYS file so that it is executed when the computer is booted. When the HIMEM utility is encountered, the program assumes control of the system's `A20 Interrupt Handler` routine. This function is part of the BIOS program and takes control of the system's A20 address line when activated.

The A20 Interrupt Handler is located at BIOS interrupt INT15 and is used to transfer data blocks of up to 64KBs in length between the system and extended memory. The INT15 function also supplies entries for the various microprocessor tables that are required for protected virtual addressing mode.

Expanded Memory (EMS)

Some publications may refer to memory above the 1MB limit as "expanded memory." However, the term `Expanded Memory` is generally reserved to describe another special memory option. In 1985, three companies (Lotus, Intel, and Microsoft) joined together to define a method of expanding the 8088's memory usage capabilities by switching banks of memory from outside the DOS memory area into the usable address ranges of the 8088. This method became known as the LIM EMS (for `Lotus, Intel, and Microsoft Expanded Memory Specification`) standard.

This idea of bank switching was not exactly new; it had been used with older computer systems before the advent of the IBM line. The LIM EMS standard simply defined how this technique should be applied to IBM PCs and their compatibles. The original standard defined specifications for both hardware and software elements of the EMS system. Figure 3.25 illustrates the basic principal behind the EMS standard.

The specification allows four 16KB areas (`pages`) of memory between C0000h and EFFFFh to be used as windows into pre-defined RAM locations above the 1MB address limit. Originally, these RAM addresses were located on special EMS RAM cards that plugged into one of the system board's expansion slot connectors. Newer system boards, based on the 80486 and Pentium microprocessors, can use their advanced virtual memory paging capabilities to handle the EMS function directly on the board.

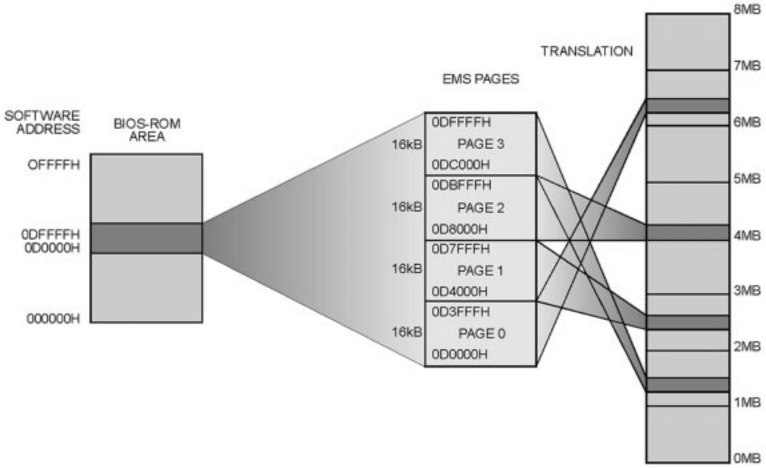


Figure 3.25

Expanded memory (EMS) operations.

Figure 3.25 depicts hex locations D0000h through DFFFFh being used as windows through which the Expanded Memory addresses are translated. In reality, the four 16KB windows can be selected from anywhere within the LIM EMS-defined address range, and can be relocated to anywhere within the 32MB physical address range.

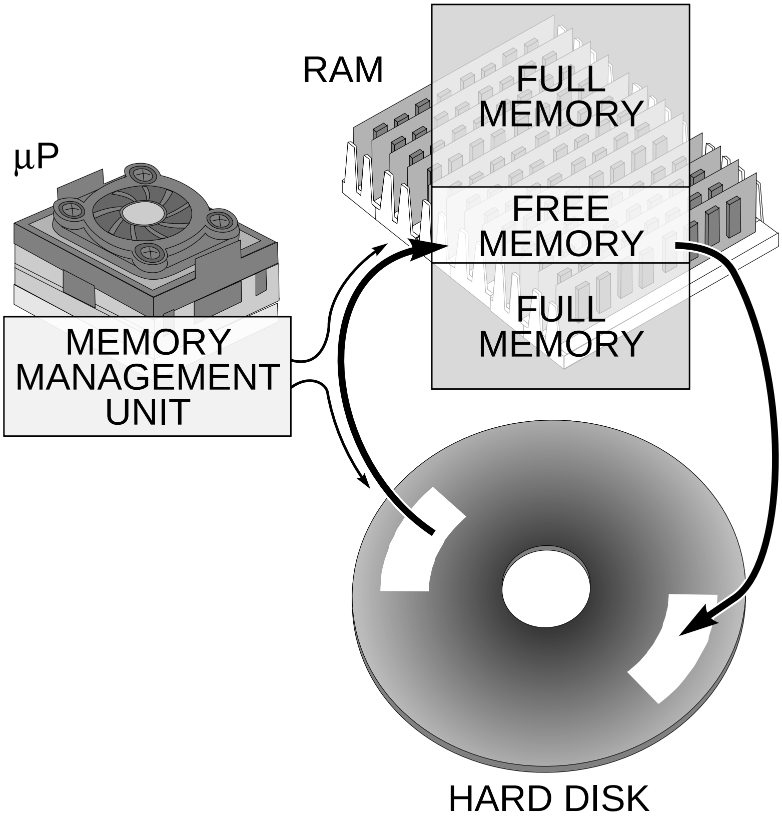
The EMS software specifications consist of predetermined programs called `Expanded Memory Manager` (EMM) drivers that work with application software to control the bank-switching operations. These drivers contain special function calls that application programs can use to manipulate the expanded memory. Note, however, that the application software must be written to take advantage of the EMS function calls. EMS versions before 4.0 made provision for the expanded memory to be used only as data storage areas. Programs could not actually be executed in these areas. Versions 4.0, and later, support much larger bank-switching operations, as well as program execution and multitasking.

Virtual Memory

The term `virtual memory` is used to describe memory that isn't what it appears to be. Virtual Memory is actually disk drive space that is manipulated to seem like RAM. Software creates virtual memory by swapping files between RAM and the disk drive, as illustrated in Figure 3.26. Because the swapping represents a major transfer of information that involves the hard-disk drive, an overall reduction in speed is encountered with virtual memory operations.

Figure 3.26

Virtual memory operations.



CONFIG.SYS

Objective

During installation, DOS versions from 5.0 forward create a system file called CONFIG.SYS. This particular filename is reserved by DOS for use with a special file that contains setup (configuration) instructions for the system. When DOS is loaded into the system, a portion of the boot-up program automatically searches in the default drive for a file named CONFIG.SYS. The commands in this file configure the DOS program for use with options devices and applications programs in the system.

The CONFIG.SYS program is responsible for: (1) setting up any memory managers being used, (2) configuring the DOS program for use with options devices and application programs, (3) loading up device-driver software, and (4) installing memory-resident programs. These activities are illustrated by the following sample CONFIG.SYS file:

```
1   Device=C:\DOS\HIMEM.SYS
    Device=C:\DOS\EMM386.EXE 1024 RAM

2   FILES=30
    BUFFERS=15
    STACKS=64,500

3   DEVICE=C:\DOS\SMARTDRV.SYS 1024
    DOS=HIGH, UMB
    DEVICEHIGH=C:\MOUSE\MOUSE.SYS
    DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 4096/a
```

Memory Managers

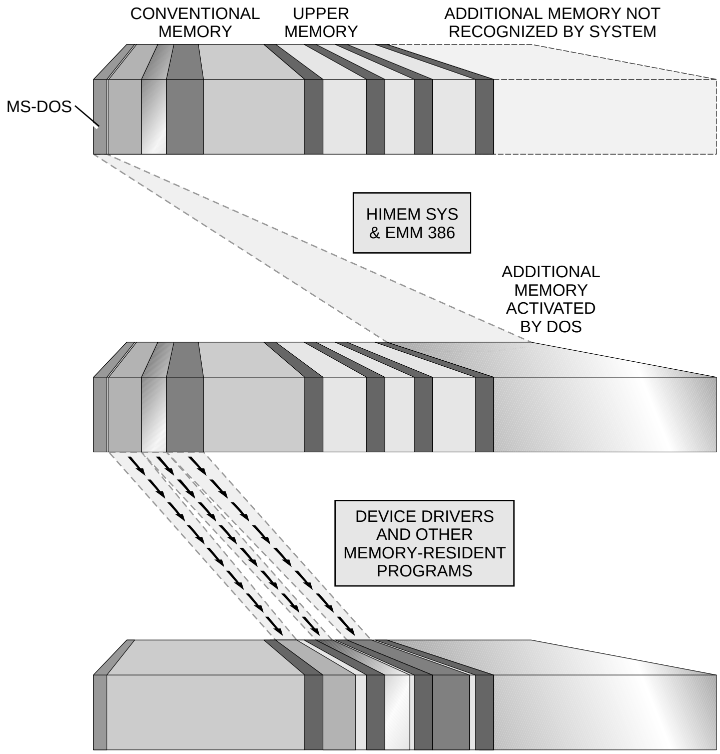
In the first section, the system's memory-manager programs are loaded. In this case, the HIMEM.SYS command loads the DOS extended memory driver. This driver manages the use of Extended Memory (XMS) installed in the system. This memory manager should normally be listed in the CONFIG.SYS file before any other memory managers or device drivers.

The EMM386.EXE program provides the system's microprocessor with access to the upper memory area. Operating together with the HIMEM.SYS program, this enables the system to conserve conventional memory by moving device drivers and memory resident programs into the UMA. This concept is described in Figure 3.27.

HIMEM.SYS also creates a 64KB area of memory just above the 1MB address space called the High Memory Area (HMA). With this, the DOS=HIGH statement is used to shift portions of DOS from conventional memory into the HMA.

Figure 3.27

Loading memory managers.



Similarly, the EMM386.EXE command loads the DOS `Expanded Memory` simulator driver. A file called LIM EMS 4.0 is another commonly used expanded memory manager that you might encounter in a CONFIG.SYS file set up for expanded memory operations.

Files, Buffers, and Stacks

In the second section of the file are the commands that define DOS for operation with optional devices and applications. The `FILES` command causes the DOS program to establish the number of files that DOS can handle at any one time at 30. This just happens to be the minimum number required to load Windows for operation. The `BUFFERS` command sets aside 15 blocks of RAM memory space for storing data being transferred to and from disks. Similarly, the `STACKS` command establishes the number and length of some special RAM memory storage operations at 64 memory stacks, with each being 500 bytes long.

Device Drivers

Device drivers are loaded in the third part of the file. Device drivers are programs that tell DOS how to control specific devices. DEVICEHIGH=C:\MOUSE\MOUSE.SYS is a command that loads a third-party device driver supporting the particular mouse being used with the system. Some device manufacturers include software Installation Utilities with the device that will automatically install its device drivers into the CONFIG.SYS (or AUTOEXEC.BAT) files during the installation process. With other devices, the device drivers must be installed by manually updating the CONFIG.SYS and AUTOEXEC.BAT files. The device's Installation Instructions will identify which method must be used to install its drivers. The order in which device drivers appear in the CONFIG.SYS file is important. The recommended order for listing device drivers is; (1) HIMEM.SYS, (2) the expanded memory manager if installed, (3) the EMM386.EXE command, and then (4) any other device drivers being used.

The `SMARTDRV.SYS` driver establishes a disk cache in an area of extended memory as a storage space for information read from the hard-disk drive. A `cache` is a special area of memory reserved to hold data and instructions recently accessed from another location. A `disk cache` holds information recently accessed from the hard-disk drive. Information stored in RAM can be accessed much more quickly than that stored on the hard drive. When a program or DOS operation requests more data, the SMARTDRV program redirects the request to check in the cache memory area to see whether the requested data is there. If SMARTDRV finds the information in the cache, it operates on it from there. If the requested information is not in the cache, the system accesses the hard drive for it.

Using this technique, the overall operating speed of the system is improved. When the system is shut down, SMARTDRV copies the most current information onto the hard drive. Therefore, no data is lost due to it being stored in RAM. The idea behind SMARTDRV operations is described by Figure 3.28.

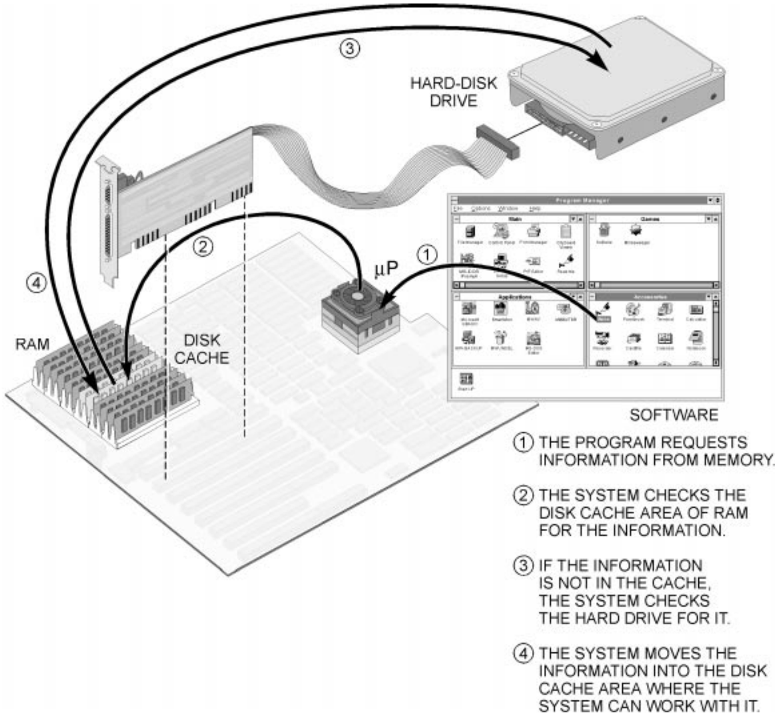


Figure 3.28

How SMARTDRV works.

The 1024 modifier establishes a memory cache size of 1MB (1024k of memory) in extended memory. This is a typical cache size for SMARTDRV; 2MB (2048k), however, is probably the most efficient size for the cache. This is because the larger the cache size, the greater the chance that the requested information is in the cache. So there is no need to go to the hard drive for the information. If the command is modified further by an /a extension, the cache is established under an expanded memory operation rather than extended memory. Extended memory is the default for SMARTDRV operations.

The `RAMDRIVE.SYS` driver simulates the organization of a hard-disk drive in RAM memory. This type of drive is called a `Virtual Disk`. In this case, the DEVICEHIGH= command loads the RAM-DRV into the upper-memory area rather than the base-memory area, where a simple DEVICE= command would run it. Likewise, the DOS=HIGH,UMB command shifts the operation of DOS into the high-memory area, and gives the application access to the upper-memory area.

The operation of both the SMARTDRV.SYS and RAMDRIVE.SYS device drivers is governed by the HIMEM.SYS memory manager. This is normal only because both programs involve the use of memory beyond the 1MB conventional-memory level. Likewise, the DEVICEHIGH= and DOS=HIGH commands that move programs into the upper memory area perform under the guidance of the HIMEM.SYS manager.

The fourth portion of the file sets the system up to use special keyboard shortcuts available in the `DOSKEY` program. DOSKEY is a type of program referred to as a `memory-resident` program. Memory-resident programs are programs that run in the background of other programs.

The DOS INSTALL command is placed in the CONFIG.SYS file to load memory-resident files into memory when DOS starts up. These files remain in memory as long as the system is on. A common install command is: INSTALL=C:\DOS\SHARE.EXE. The SHARE program provides the capability to share files in a networked, or multitasking, environment.

Other common CONFIG.SYS commands include:

- BREAK
- COUNTRY
- DRIVPARM
- LASTDRIVE
- NUMLOCK
- REM
- SET
- SHELL
- INCLUDE
- MENUCOLOR
- MENUDEFAULT
- SUBMENU

The definitions and usage of these commands are covered in detail in the MS-DOS User's Guide. The DOS installable device drivers are also defined in that publication.

Altering CONFIG.SYS Steps

The operation of the CONFIG.SYS file can be altered or bypassed by pressing selected keyboard keys during the bootup process. Holding the SHIFT key, or pressing the F5 key while the MS-DOS message "Starting DOS..." is on the screen, causes the bootup process to skip all the commands in the CONFIG.SYS file. This action also bypasses all the steps of the AUTOEXEC.BAT file (discussed in the next section).

When this option is used, the system boots up with a complete set of default settings. No installable device drivers are installed, the current directory is set to C:\DOS, and you may receive a "Bad or missing command interpreter" message. If this message is received, the system asks you to manually enter the path to the COMMAND.COM file.

Objective

Similarly, pressing the F8 function key while the DOS message is on the screen causes the system to stop between each CONFIG.SYS command, and ask for verification before proceeding. This can be very helpful in troubleshooting configuration and bootup problems. This action also causes the system to ask the user whether the AUTOEXEC.BAT file should be run or skipped. Placing a question mark after a CONFIG.SYS command (before the = sign) causes the system to automatically seek verification whenever the system is booted up.

DOS comes with several other standard device driver programs. These include:

- KEYBOARD.SYS
- DISPLAY.SYS
- ANSI.SYS
- DRIVER.SYS
- PRINTER.SYS

KEYBOARD.SYS is the DOS default keyboard definition file. The DISPLAY.SYS driver supports code-page switching for the monitor type in use by the system. A `code page` is the set of 256 characters that DOS can handle at one time, when displaying, printing, and manipulating text. ANSI.SYS supports ANSI escape-code sequences used to modify the function of the system's display and keyboard. This file is also required to display colors on the monitor in DOS. DRIVER.SYS creates the logical drive assignments for the system (such as A: and C:). Finally, the PRINTER.SYS driver supports code-page switching for parallel ports. All these drivers are normally found in the DOS directory.

POWER.EXE

A special, power-saving program called POWER.EXE is designed for use in notebook computers. When it is loaded in the last line of the CONFIG.SYS file, and the system hardware meets the Advanced Power Management specification, the power savings can be as high as 25%. This is an important savings when you are discussing the operation of a battery, and its length of operation before it needs to be recharged. The POWER.EXE file must be available in the C:\DOS directory.

AUTOEXEC.BAT

After completing the CONFIG.SYS operation, DOS searches for the presence of a file called the AUTOEXEC.BAT file. This file contains a `batch` of DOS commands that are automatically carried out when DOS is loaded into the system. This file can also be re-executed from the DOS prompt if you simply type the command "AUTOEXEC." This is not true of the CONFIG.SYS file however. The system must be restarted to perform the commands in this file.

Refer to the following sample AUTOEXEC.BAT file:

```
DATE
TIME
PROMPT=$P$G
SET TEMP=C:\TEMP
PATH=C:\;C:\DOS;C:\MOUSE
DOSKEYSMARTDRV.EXE 2048 1024
CD\
DIR
```

The first two commands cause DOS to prompt you for the date and time (because DOS does not automatically do this when an AUTOEXEC.BAT file is present). The `PROMPT=$P$G` command causes the active drive and directory path to be displayed on the command line. The `SET TEMP=` line sets up an area for holding data temporarily in a directory named TEMP.

The `PATH` command creates a specific set of paths that DOS is to use to search for program files. In this case, DOS searches for executable files first in the root directory, followed by the DOS directory, and finally through the MOUSE directory. This statement effectively lets the Mouse program be executed from anywhere in the system. Upon receiving the command, the operating system looks through all of the directories in the path until it finds the specified filename.

The `syntax` (punctuation and organization) of the PATH command is very important. Each entry must be complete from the root directory and must be separated from the previous entry by a semicolon. There should be no spaces in the PATH command.

The `DOSKEY` command loads the DOSKEY program into memory. Following this, the SMARTDRV.EXE 2048 1024 command configures the system for a 2MB disk cache in DOS and a 1MB cache for Windows. After the cache has been established, the CD\ command causes the DOS default directory to change to the root directory. The last line causes a DOS DIR command to be performed automatically at the end of the operation.

The execution of the AUTOEXEC.BAT file can be interrupted by pressing the `Pause` key on the keyboard. The program can be restarted by pressing any key. With DOS version 6.2, the F8 interactive bypass procedure, described for use with the CONFIG.SYS file, was extended to include the AUTOEXEC.BAT file.

You can use the DOS batch file commands to construct elaborate startup procedures. Other programs designed to test ports and peripherals can be constructed using these commands. These test files can be named using the DOS filename conventions. They must be stored with a .BAT extension to be executable from the DOS prompt, but the extension does not need to be entered in order for the program to run.

Neither of these two special files are required for normal operation of the computer with DOS. However, they can prove to be very useful in tailoring the operation of the system to your personal use, or to the requirements of different software applications packages. To determine whether either of these files already exist on your DOS disk, simply type the DIR command at the DOS prompt (with the disk in the default drive).

Refer to the MS-DOS® User's Guide manual for more information about the creation and use of the CONFIG.SYS and AUTOEXEC.BAT files. Other DOS utilities for disk management are covered in the disk drive and preventive maintenance chapters (Chapters 8 & 13).

DOS Editor

Later versions of DOS contain a small text editor program that enables users to easily modify text files. This package is started by typing `EDIT` and the filename at the DOS prompt. The DOS editor's working screen appears. The editor is particularly useful in modifying the CONFIG.SYS and AUTOEXEC.BAT files. The DOS editor is an editor for unformatted text files. It does not introduce formatting codes, such as underlining and italics, into the text in the manner that more powerful word processors do. This is an important consideration when dealing with DOS utility files. Formatting codes can introduce errors in these files, because DOS is not able to recognize them.

DOS Versions

Although DOS has remained compatible with its original design, this does not mean that it has not changed significantly since its original version.

In July of 1981, Microsoft purchased the rights to a personal computer DOS from the Seattle Computer Products and promptly named it MS-DOS. A month later IBM began shipping a private labeled version of the Microsoft package that it named PC-DOS 1.0.

In May of 1982, Microsoft released MS-DOS version 1.1 to IBM for its units, and released its own brand name DOS product, MS-DOS 1.25, for PC-compatible computers. This version added support for 360KB double-sided floppy-disk drives.

In March of 1983, MS-DOS 2.0 was announced. It added support for 10MB hard drives, a directory tree structure, and 360KB floppy drive support to the operating system. A minor revision, titled MS-DOS 2.11, added foreign language and date features to the operating system in March of 1984.

Version 3.0 of MS-DOS was released in August of 1984. It was released along with IBM's AT model and added support for 1.2MB floppy-disk drives and larger hard-disk drives. In November of the same year, version 3.1 added support for Microsoft networks. By January of 1986, version 3.2 had entered the market and brought support for 3-1/2 inch, 720KB floppy-disk drives to the operating system.

In August of 1987, version 3.3 delivered 1.44MB floppy-disk drive and multiple 32MB partitions for hard drives.

Version 4.0 of MS-DOS was released in June of 1988. This new version introduced a graphical shell for DOS, a mouse interface, and expanded memory drivers. By November, version 4.01 was being shipped to clean up problems with the 4.0 version.

The next version of MS-DOS didn't appear until June of 1991. Version 5.0 brought a full screen editor, task-swapping capabilities, `Undelete` and `Unformat` commands, and QBASIC to the operating system. In addition, it included support for upper memory blocks, larger hard disk partition sizes (up to 2GB), support for 2.88MB floppies, and the capability to load DOS into the HMA, as well as to load device drivers into the UMBs.

Microsoft began shipping the 6.0 version of MS-DOS in March of 1993. This new version included a DoubleSpace disk compression utility that enabled users to double the storage capacity of their hard-disk drives. Over 1 million copies of this version sold within the first month and a half. An enhanced version, MS-DOS 6.2, was released in November of the same year. By February of 1994, legal problems over the compression utility caused Microsoft to release version 6.21 with the utility removed. However, by June, version 6.22 appeared with the compression software back in the operating system under the name DriveSpace.

In April of 1994, IBM released a new version of PC-DOS. This was version 6.3. In January of 1995, they followed with PC-DOS 7, which included data compression for hard disk doubling. This marked the last release of a major command line-based DOS operating system. Table 3.2 summarizes the development of DOS products.

Table 3.2

DOS Development Timeline

Year	Version	Features
1981	V1.0	First operating system for IBM-PC.
	V1.25	Double-sided disk support and bug fixes added, widely distributed by OEM's.
1983	V2.0	Hierarchical file support and hard disk support.
	V2.01	International support added.
	V2.11	V2.01 with bug fixes.
1984	V3.0	Introduced with AT model. Support for 1.2MB floppy, and larger hard disk sizes.
	V3.1	Support for Microsoft Networks added.
1986	V3.2	Enhanced support for new media types added.
1987	V3.3	Support for 1.44MB floppy, support for 4 serial ports, hard disk partitions > 32MB, improved national language support.
	V4.0	Dosshell, support for TSR's, expanded memory drivers, install program (select), mem command.
	V4.01	V4.0 with bug fixes.

1992	V5.0	support for upper memory blocks, larger partition sizes (< 2GB), loading device drivers in UMB, improved dosshell and on-line help, support for 2.88MB drives, Qbasic, improved system editor (edit).
1994	V6.0	DoubleSpace Disk compression introduced.
	V6.22	DriveSpace Disk compression, replaces DoubleSpace.

Installing DOS

Objective

In the earliest versions of DOS, the operating system was contained on two disks, the System Disk and the Supplemental Disk. Because the early PCs operated from floppy-disk drives, they booted directly to the system disk. The most-used DOS utility functions were loaded into RAM. For advanced DOS functions, the Supplemental disk was inserted in the drive when requested by the system.

When the PCs moved to hard drive operations, the main DOS files were placed on the hard disk as part of its formatting process. The other DOS files were typically copied into a C:\DOS directory when the unit was set up.

The installation of newer DOS versions (after 5.0) is a relatively easy process. It is controlled by the file `SETUP.EXE`. The only response required by this program is to tell it whether DOS should be installed on the hard-disk drive. The installation process is so automated in newer versions that it runs the Setup.exe program automatically on new systems. This is accomplished simply by starting the system with the first DOS disk (1 of 3) in the floppy drive.

The setup program's installation screen defaults to the C:\DOS sub-directory to install the DOS files. The files on the installation floppies are compressed, so they cannot simply be copied onto the hard drive. As the DOS files are installed, the setup program automatically determines the amount of memory, the number of drives, and other configuration information about your computer.

With versions of MS-DOS from 5.0 forward, the setup program also creates the files `AUTOEXEC.BAT` and `CONFIG.SYS`. Once setup has determined the configuration, the findings are presented for the user to check. If the setup program incorrectly determines the configuration, it can be changed during the setup.

DOS Shell

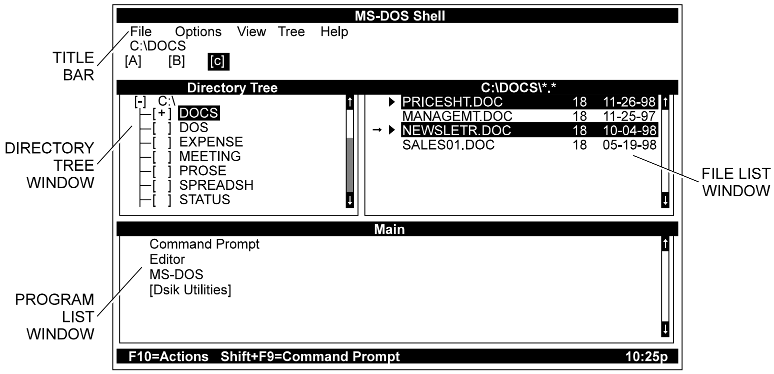
Versions of DOS from 4.0 forward offer an option to install the `DOS shell`. The DOS shell is a multiple window user-interface that simplifies performing DOS operations. The DOS shell interface is depicted in Figure 3.29.

As illustrated in the figure, the DOS shell divides the screen into four basic areas:

- Title bar
- Directory tree window
- File list window
- Program list window

Figure 3.29

The DOS shell.



The drive icons under the title bar are used to select the disk drive in which to carry out any DOS functions. The drive icon is selected by highlighting it with the cursor and then pressing the Enter key. When the drive is selected, its directory structure appears in the directory tree window.

The directory tree window shows the organization of the selected disk's directory structure. As with drive selection, a particular directory or sub-directory can be selected using the cursor. When this is done, the contents of the directory appear in the file list window.

The program list window shows programs that can be run directly from the shell such as QBASIC and DOS Editor.

Beginning with version 4.0 MS-DOS enabled `Task Swapping`. By enabling the Task Swapper utility in the Title Bar's Options menu, multiple programs could be run simultaneously. The execution of the programs was not simultaneous, but there was no need to exit from one program to start another. All the active programs appear in the Active Task window. To switch between them is a simple operation of getting to the shell and clicking on the desired program in the window.

The shell makes it easy to locate and start files on the disk. Any program can be located by scrolling through the different windows. When the desired file is located, it can be run by simply highlighting it with the cursor and pressing the Enter key, or by clicking a mouse button. There are no cryptic commands to remember and no long paths to type.

Summary

This chapter has looked at operating systems in depth. It has concentrated on the ROM BIOS and disk operating system. At this point, you should be able to discuss the duties of the BIOS and disk operating system. You should also be able to describe the events involved in booting up a computer to the operating system. In addition, you should be able to describe the general use of the disk operating system after it has been loaded. The next chapter examines the Microsoft Windows operating environment and the Windows 95 Operating System. These systems differ from the disk operating system that we have covered in this chapter in that they were designed to be graphical in nature and use.

Lab Exercise

A hands-on lab procedure corresponds to the theory materials presented in this chapter. Refer to the lab manual and perform Procedure 2, "Introduction to DOS."

Review Questions

1. What function does the POWER.EXE file perform and where is it commonly found?
2. Under what conditions is it normally necessary to run the `CMOS Configuration Setup` program?
3. What type of operating system breaks the tasks associated with a process into its various threads for execution?
4. Which DOS command prepares a disk to function as a self-booting disk?
5. Describe the DOS prompt.
6. In terms of managing processes, what type of operating system is DOS?
7. For which file is the operating system loader looking during the bootup process?
8. What advantage does the DOS shell offer to the system?
9. What is the purpose of placing a SMARTDRV.EXE command in the CONFIG.SYS file?
10. What is the first event that occurs when the system is turned on or reset?
11. Which memory manager should always be listed before any other memory managers or device drivers?
12. If the system locks up while the message "Starting DOS" is on the screen, what two actions should be performed?

13. Pressing the F8 key while the "Starting DOS" message is on the screen has what effect on the system?
14. Where do you find the system's memory managers listed?
15. From the system startup point of view, how do a cold and a warm boot differ?
16. What does the "*" character stand for when used in a DOS filename?
17. How does the XCOPY command differ from the copy or diskcopy command?
18. List the three files that must be located in the root directory in order for you to successfully boot MS-DOS.
19. How are multi-tasking and multi-user systems different?
20. Write a DOS command that can be inserted in the AUTOEXEC.BAT file to cause the active path and directory to be shown on the DOS command line.
21. Which filename extensions enable programs to be started directly from the DOS prompt?
22. What does HIMEM.SYS do?
23. Write a DOS command that can be used to make a duplicate of another disk.
24. What condition is indicated by the presence of an A:\> prompt on the monitor screen?
25. What does the BUFFERS= command in the CONFIG.SYS file do?

Review Answers

1. POWER.EXE is a special, power-saving program designed for use in notebook computers. When it is loaded in the last line of the CONFIG.SYS file, it can provide savings as high as 25%. For more information, see the section "POWER.EXE."

2. When setting up a computer for the first time, or when adding new options to the system. For more information, see the section "CMOS Setup Routines."

3. Multiple-process operating systems. For more information, see the section "Operating Systems."

4. Format /s. For more information, see the section "Drives and Disks."

5. The DOS prompt is a screen symbol that marks a point where DOS commands can be entered. The prompt includes the designation of the active disk drive, and may contain directory and subdirectory path information (for example, C:\dos\mouse). For more information, see the section "DOS Command Line."

6. DOS is a single-process operating system that operates in `interactive` mode. For more information, see the section "Operating Systems."

7. The BIOS begins looking through the system for a Master Boot Record. For more information, see the section "Boot-up."

8. The DOS shell is a multiple-window user interface that simplifies performing DOS operations. For more information, see the section "DOS Shell."

9. To establish a disk cache in the extended memory area for data read from the hard drive. For more information, see the section "Device Drivers."

10. The system is reset to its starting condition. For more information, see the section "Post Tests and Initialization."

11. The HIMEM.SYS command. For more information, see the section "Memory Managers."

12. Press the F5 key to skip the CONFIG.SYS and AUTOEXEC.BAT portions of the bootup to see whether information in these files is causing conflicts. If the system boots up without these files, reboot and use the F8 key to

single-step through the two files until the conflict is located . For more information, see the section "Altering CONFIG.SYS Steps."

13. It causes the system to move through the CONFIG.SYS and AUTOEXEC.BAT files one command at a time. For more information, see the section "Altering CONFIG.SYS Steps."

14. In the system's CONFIG.SYS file. For more information, see the section "CONFIG.SYS."

15. In a cold boot situation, the system executes the entire BIOS startup routine. In a warm boot, only some of the POST tests are performed on the system. This makes a warm boot much quicker. For more information, see the section "Initial POST Checks."

16. The "*" symbol is used as wild card character that can be substituted for an unknown character or character string in DOS commands. For more information, see the section "DOS Shortcuts."

17. The XCOPY command copies all the files in a directory, along with any sub-directories and their files; the file copy command copies a specified file from one place (disk or directory) to another. For more information, see the section "Files and Filenames."

18. MSDOS.SYS, IO.SYS, and COMMAND.COM. For more information, see the section "Bootup."

19. The multi-user system switches between different users at multiple locations; multi-tasking systems switch between different applications at a single location. For more information, see the section "Operating Systems."

20. PROMPT=PG. For more information, see the section "Directories."

21. .EXE, .COM, and .BAT For more information, see the section "DOS Command Line."

22. It governs the use of extended memory. For more information, see the section "Extended Memory."

23. DISKCOPY A: For more information, see the section "Drives and Disks."

24. DOS is operational and the currently active drive is the A: floppy drive. For more information, see the section "Bootup."

25. It sets aside blocks of RAM memory space for storing data being transferred to and from disks. For more information, see the section "Files, Buffers, and Stacks."

Chapter 4

Microsoft Windows

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Describe the development of the Windows software program
- List the core files in the Windows 3.x and Windows 95 structures
- List the standard INI files in Windows 3.x and describe their functions
- Describe advances brought to the Windows package by the 3.11 version
- State reasons for implementing security measures in multi-user or networked systems
- Describe the bootup sequence employed by Windows 95
- Bypass and correct inoperable Windows startup sequences
- Describe the major files of the Windows 3.x and Windows 95 operating systems
- Describe multitasking operations in Windows 3.x and Windows 95
- State major differences between Windows 3.x and Windows 95

continues

- Use Windows 3.x to find, copy, rename, delete, move, and recover files
- Alter typical Windows 3.x Control Panel values
- Use the various Windows setup modes to install Windows into a system
- Manipulate the Windows 95 Registry structure
- Use the Policy Editor to change Windows policy settings
- Create an Emergency Startup Disk for Windows 95
- Use various Safe mode startup scenarios
- Load driver software for devices added to the system
- Use both Windows 3.x and Windows 95 to create, delete, and navigate through directories
- Start Windows in various operating modes

Introduction

The preceding chapter explored operating systems in some detail. Most of the information covered BIOS operations and DOS command-line structure. Interestingly, the chapter concluded with a discussion of a graphical user interface for MS-DOS—the DOS shell. It may be easy to think that the Windows programs evolved out of the DOS program. This is not the case, however. Microsoft Windows was first introduced in 1984, 10 years before Microsoft introduced its final version of MS-DOS. The pressure for a GUI operating system for PCs actually came from the Apple operating systems.

This chapter continues with operating system software by examining Microsoft's Windows programs. The first half of the chapter deals with the 3.x versions of Windows. In these versions, Windows operates as an `operating environment`. This can easily be thought of as a three-layer operating system—the BIOS, DOS, and Windows.

The final sections of the chapter deal with Windows 95. Unlike its predecessors, Windows 95 is a true operating system. No DOS foundation is required.

Windows Evolution

Objectives

In April of 1983, Microsoft demonstrated a graphics interface manager that would later become Windows. It gave the appearance of overlapping window panes with various programs running in each window. In November of the same year, Microsoft formally announced Windows and set a release date of April 1984. Interestingly, IBM passed on the opportunity to market Windows with its units three different times. They were busy developing a GUI called TopView for their systems.

Microsoft announced Windows 1.0 in June of 1985 and began shipping in November. It found a PC market that was steeped in command-line operations. Many industry analysts predicted it would come and go and "real computer users" would hold on to their DOS disks.

Version 2.0 was announced in April of 1987 and actually hit the market in October, along with a version called Windows/386. By December of that year, Microsoft had shipped over one million copies of Windows. Two versions of Windows 2.1 shipped in June of 1988 under the titles Windows/286 and /386.

Windows 3.0 did not make it to the market until May of 1990. It opened with a $3 million dollar, first-day advertising campaign. In March of 1992, Microsoft produced its first television advertising campaign for the upcoming Windows 3.1 version. The new version began shipping in April and had reached a level of one million copies shipped by June. The 3.1 version migrated into Windows for Workgroups (WfW) in November of 1992. This version integrated peer-to-peer networking directly into the operating environment. It was quick and easy to install and set up a workgroup network to share information and resources among different computers. These were terms that were not usually associated with networking computers together. By April of 1993 the number of Windows units sold had risen to 25 million licensed copies. By October, Microsoft had begun shipping 3.11, the final 3.x version of Windows.

In September of 1994, Microsoft announced the next version of Windows. They called it Windows 95. The first version was released in August of 1995 and sold one million copies during the first week on the market. Within a month, the sales of Windows 95 climbed to over seven million copies, and by March of 1996 had topped 30 million copies.

From this historical outline, it is easy to see why this chapter focuses on Windows 3.x and Windows 95. Because of the huge installed base of these products, any technician needs to be familiar with these operating systems for the immediate future (see Table 4.1).

Table 4.1

Windows Development Timeline

Year	Version	Features
1983		Graphics interface manager demonstrated
1985	V1.0	First official Windows release
1987	V2.0	Task switching of applications
	Windows/286	Use of all the extended memory for applications
	Windows/386	Cooperative multitasking of applications
1990	V3.0	Preemptive multitasking of applications, enhanced memory support, use of icons, Program Manager interface
1992	V3.1	(Windows for Workgroups) Integrated peer-to-peer networking directly into the operating environment
1993	V3.11	Upgraded 32-bit software and disk capabilities, BIOS calls removed from file accesses
1995	Windows95	Improved multimedia support, Plug and Play hardware support, 32-bit advanced multitasking function, improved email and fax capabilities, WAN usage

Microsoft Windows 3.x

Windows is a `graphical working environment` that makes it easier for people to use computers. Under Windows, each application runs in an onscreen window that can open and close. It is possible to run multiple applications simultaneously—each in its own window—and to switch easily between them. Windows can be installed on any computer using a 286 or higher microprocessor.

Windows works most effectively with the pointing device known as a `mouse`. The mouse is used to select graphical program options from the screen to be executed. It works with Windows software by moving a screen symbol, called a `cursor`, across the screen as the mouse is rolled across a desktop.

When the cursor is pointing to an `icon` (graphical symbol) onscreen, the program represented by the icon can be selected by just pressing (clicking) the left button on the mouse. Likewise, the program can be run by double-clicking the mouse button on the icon. Remember, a single click selects and a double-click executes. The icon images are usually created by the software manufacturer, and are different for each application. However, the same icon can be used to represent different programs within the system, or within a single window.

Moving the mouse in a given direction produces corresponding movement of the cursor onscreen. To select an object onscreen, just move the cursor over the item, as illustrated in Figure 4.1, and `click` one of the mouse buttons.

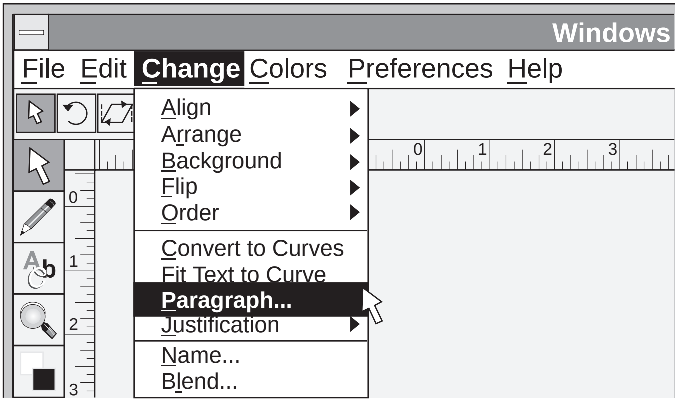


Figure 4.1

Selecting an item from the screen.

Installing Windows 3.x

Windows 3.x is relatively simple to install. A standard setup involves placing the Setup Disk 1 in the A: drive, and typing **SETUP** at the A:\>> prompt. The SETUP program examines the system to determine what hardware is being used, installs the Windows files on the hard disk, configures its initialization files with default information, searches for existing application programs on the hard disk, creates `program information files` (PIFs) about any DOS-based files it finds, and modifies the existing CONFIG.SYS and AUTOEXEC.BAT files.

When installing Windows 3.x on a system, it is important to examine the requirements of any terminate-and-stay-resident (TSR) programs that may be on the drive. These programs will remain on the drive after the installation, and could create conflicts with Windows components.

The command SETUP/A can be used to install a shared copy of Windows on a network drive. Likewise, the command SETUP/I sets up Windows to start, without regards to any automatic hardware detection.

Windows 3.1 requires a minimum of a 286 system, with 2 MB of RAM, to start in Standard mode. This includes 256 KB of free conventional memory and 192 KB of free extended memory. A memory manager such as HIMEM.SYS must also be loaded before starting Windows. Windows 3.11 requires a 386 system with a total of 4 MB of RAM to start in Standard mode.

To start Windows in 386 Enhanced mode, a minimum of an 80386SX microprocessor, 256 KB of free conventional memory, and an extended memory of 1024 KB are required. A memory manager must also be loaded ahead of Windows.

Windows checks the system memory before deciding which mode to start in. If it finds less than 1 MB of extended memory present, it automatically defaults to Standard mode. If the system possesses an 80386 microprocessor and 2 MB of RAM, it can run efficiently in 386 Enhanced mode. This mode makes use of the virtual memory capabilities of 80386 and newer microprocessors.

Running Windows

Objectives

The Windows program can be loaded into the system by typing **WIN** at the C:\>> prompt. This command executes the WIN.COM program in the Windows directory. The WIN.COM file checks the system type, memory arrangements, and device drivers to determine which mode to start in. Typing **WIN/S** forces the system to start Windows in Standard mode. Conversely, typing **WIN/3** forces the system to run in Enhanced 386 mode, if possible. WIN.COM uses the file DOSX.EXE to start in Standard mode, and WIN386.EXE to start in Enhanced mode. An extended memory manager (XMS) such as HIMEM.SYS, QEMM, or 386-MAX is required before either mode can be used.

`WIN/S` could be considered 80286 mode. The 80286 microprocessor did not have the capability to run separate applications simultaneously and keep them from interfering with each other. Although it was possible to task-switch in Standard mode, multitasking was not possible.

The Windows 386 Enhanced mode was designed to take advantage of the 80386 microprocessor's `virtual protected mode`. In this mode, the 80386 can act like multiple 8086 microprocessors operating simultaneously. This is also known as `virtual 86 mode`. Each virtual 86 environment established is known as a `virtual machine` and operates like a completely independent 8086 machine.

This mode is extremely useful for running DOS applications from the Windows environment—a condition discussed in greater detail later in section titled "DOS and Windows."

In any event, WIN.COM brings together the Windows core files, driver files, fonts, and support files for non-Windows applications, and DOS-mode–specific programs.

The core files are as follows:

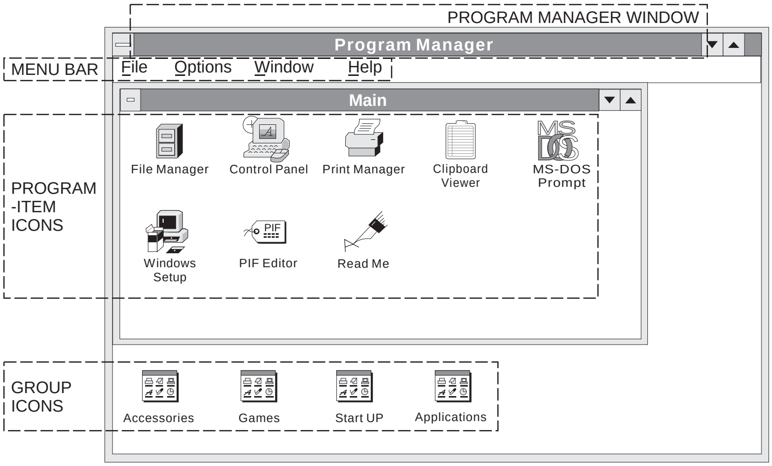
- KRNL386.EXE (Windows Kernel)
- USER.EXE
- GDI.EXE

The Kernel is the file that controls the usage of the system's memory and I/O resources. It also loads and executes Windows applications. The USER.EXE file creates and controls the various onscreen windows. It is responsible for handling requests to create, move, size, or delete windows onscreen. In addition, the USER.EXE file controls user input and output through the mouse, keyboard, and communications ports. The GDI file controls the Windows GUI functions. These files are located in the `\Windows\System` directory created when Windows was installed.

The driver files provide the hardware-specific interface between the Windows software and the physical devices in the system. What appears onscreen next is the `Program Manager`, depicted in Figure 4.2. The Program Manager is the application coordinator used to `associate` related software applications and data into groups. It uses icons to quickly locate and run applications that you use. On the same screen as the Program Manager are the different groups of applications. Each group has its own icon to represent that group of applications. The default groups included when Windows is installed are Main, Startup, Accessories, Games, and Applications.

Figure 4.2

Windows Program Manager.



The Main group includes system applications that control files, control printing, set up peripherals, customize the appearance of Windows (called the desktop), and manage files and disks. The Startup group includes optional applications that Windows activates immediately on startup. The Accessories group includes desktop accessories such as a simple word processor, calculator, calendar, card file, note pad, and a painting program. The Games group includes two games, Solitaire and Minesweeper. The Applications group includes programs that Windows found when it was installed, such as Backup, and anti-virus programs that run under both DOS and Windows.

The Windows program can be operated from the keyboard, but it was designed to be used with the mouse. When the mouse is used, a pointer is produced onscreen. By moving the mouse, which moves the pointer over a selection, and pressing the left mouse button twice in rapid succession (called double-clicking), the selection is chosen. The item can be dragged across the screen by moving the pointer over it, holding down the mouse button, and then moving the mouse. Then, release the mouse button to drop the item.

Main Window

The Main group, depicted in Figure 4.3, contains the File Manager, Control Panel, Print Manager, Clipboard Viewer, MS-DOS Prompt, and Windows Setup icons.

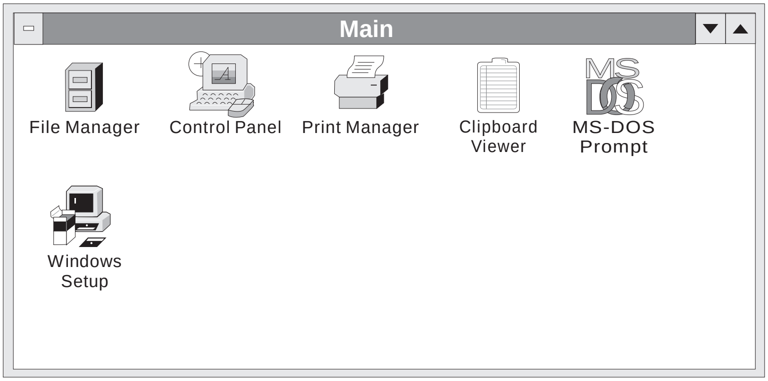


Figure 4.3

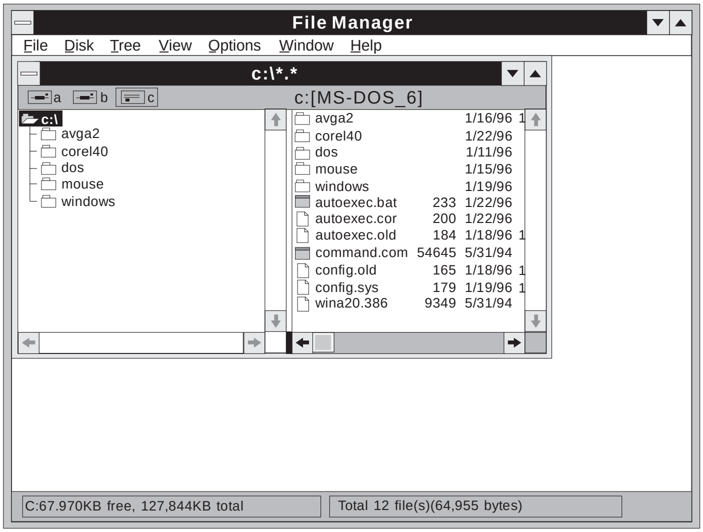
The Main window.

File Manager

The File Manager is an application used to manage files and disks. It enables the user to copy, move, and delete files on any of the system's drives. Its icon resembles a file cabinet. The File Manager's screen is divided into three parts. The bar at the top of the screen with File, Disk, Tree, View, Options, Window, and Help options, is called the `menu bar`. The left side shows a directory tree, with one directory or subdirectory highlighted to show that it is selected. The right side of the screen shows the files contained in the selected directory or subdirectory. At the bottom of the screen is the `status bar` that shows the number of files and number of bytes the selected directory consumes. This is illustrated in Figure 4.4.

Figure 4.4

The File Manager screen.



Multiple directories can be displayed on the same screen, making it easier to perform file operations. Files can be moved or copied from one directory to another. In a move operation, selected files are picked up from one directory or disk and deposited in another directory or disk. In a copy operation, the file is deposited in the new directory, and remains in the original directory.

Multiple files can be selected for move or copy operations. To select sequential files from the list, click on the first file. Then press Shift, and click on the last file. All the files between and including the first and last file will be selected. To select nonsequential files for file operations, press the Ctrl key and click on each filename. After the desired files have been selected, a move, copy, or delete operation can be performed on all the highlighted files.

The File Manager can perform other DOS-like file management functions. These functions include disk formatting, as well as directory and file organization and management. Under the Disk entry on the menu bar, options to Format, Label, and Copy disks are available. Self-booting system disks can also be created from this menu.

Under the File option, the capability to Open or Run selected files is presented. It is also possible to create new directories, search selected directories for a given file, and sort files by their attributes from this menu.

The correct procedure for installing a program, or running it from the File Manager window is as follows:

1. Click on the drop-down File option on the menu bar.
2. Click on the Run entry.
3. Enter the filename of the program to be installed, or run.

These same steps can be used to start programs from the menu bars of other windows.

Programs can also be set up to automatically start along with Windows. This operation can be accomplished by placing the program's icon in the Startup window, or by placing a `RUN=programname` or `LOAD=programname` line in the WIN.INI file.

The View menu option enables the user to control how files and directories appear in the File Manager window. Files can be sorted by name, type, size, and date of creation through this menu. Other options in this menu control how the directory tree is displayed onscreen.

Control Panel

Objectives

The `Control Panel`, depicted in Figure 4.5, enables users to modify the system's date and time, and to determine how Windows responds to the mouse and keyboard. Its options can be accessed by double-clicking on its icon in the Main window. The Control Panel also facilitates the installation of the system printer(s) and customizing of the desktop, as well as featuring Color, Desktop, Fonts, International, and Sound options that can be used to customize Windows to the user's preference.

Customizing the Windows desktop from the Control Panel involves setting screen colors and features such as the wallpaper.

Wallpaper is the picture displayed behind the Program Manager window.

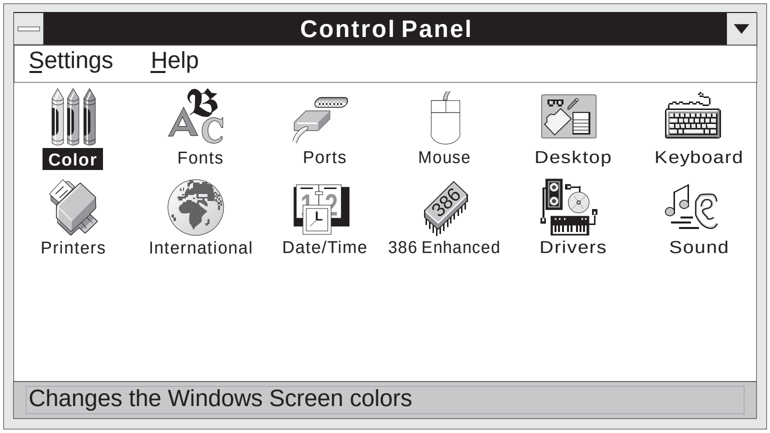


Figure 4.5

The Windows Control Panel.

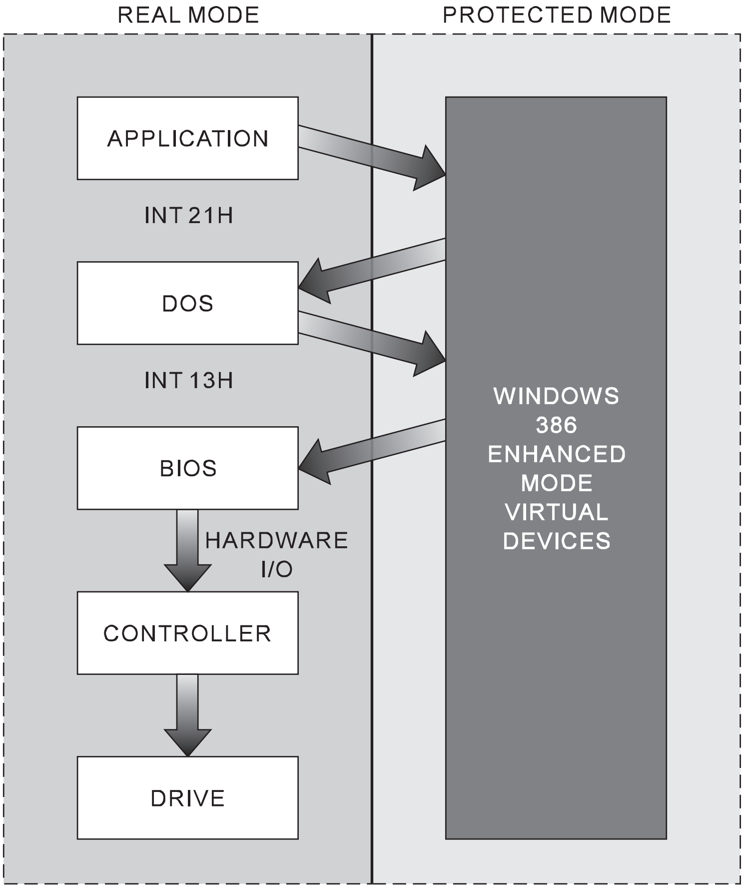
Still other options, such as Ports and Drivers, are used to further configure the operation of Windows. The `Ports` option is used to set up the serial ports of the computer. The `Drivers` option is used to install software drivers for other purposes such as sound boards, mice, and CD-ROM drives.

Windows 3.1 introduced 32-bit access to the Windows package. This feature can be found in the Control Panel window under the Virtual Memory option of the 386 Enhanced icon. Contrary to the sound of its name, 32-bit access has nothing to do with moving data in 32-bit blocks. Instead, 32-bit access is a method of reducing the need to move back-and-forth between real, and protected memory modes, when an access request is made.

Before version 3.1 came out, a simple request for a hard disk access while running in protected mode would result in Windows, DOS, and the BIOS handing the request back-and-forth a number of times—that is, application-to-Windows-to-DOS-to-Windows-to-BIOS-to-Windows-to-DOS-to-Windows-to-application. Figure 4.6 shows such an operation.

Figure 4.6

The Windows/DOS/BIOS relationship.



In Windows 3.11, the work associated with this type of operation was reduced by removing the BIOS completely from the loop. A protected-mode device driver called FastDisk emulates the BIOS routines in Windows. The upgraded 32-bit software capabilities in 3.11 sped up the operation of the system by eliminating the changes between real and protected modes that occurred each time Windows had to hand over control to the DOS or BIOS. Additional speed was obtained due to the reduced information processing required, by cutting the BIOS portion out of the loop. Figure 4.7. depicts the advanced 32-bit access process using FastDisk.

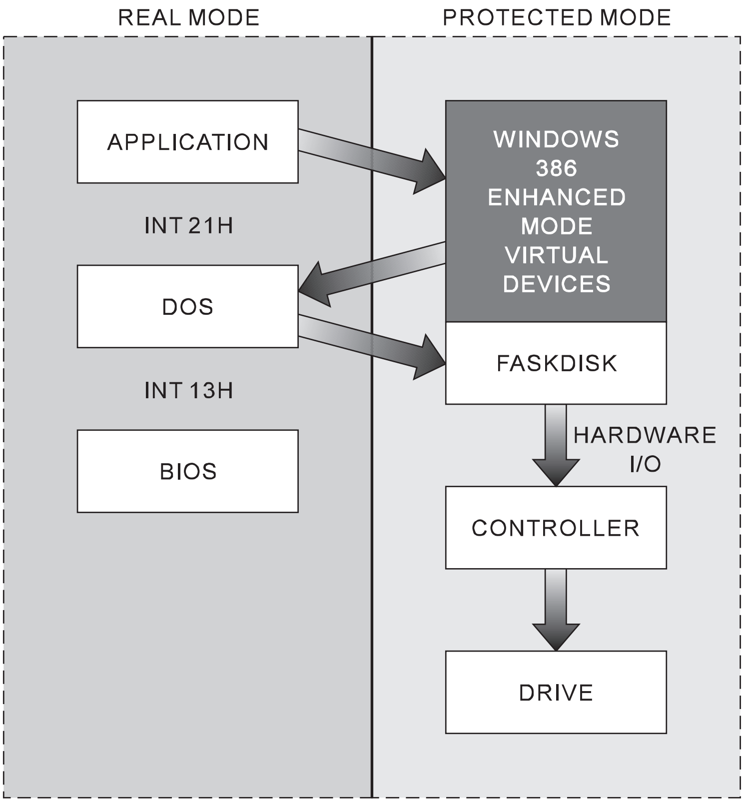


Figure 4.7

32-bit access with FastDisk.

Windows 3.11 adds 32-bit access capabilities to file accesses as well as disk accesses. This further increases the system's overall operating speed by removing BIOS calls from file accesses as well.

Windows establishes a swap file on the hard disk to free up memory and speed up operation in 386 Enhanced mode. This file reserves space on the hard drive to allow data to be swapped back-and-forth between the drive and system memory.

A temporary swap file named WIN386.SWP is created when Windows starts, and disappears when Windows is shut down. If possible, Windows will create a permanent swap file through its Setup program. This file is governed by two files located on the HDD: SPART.PAR and 386SPART.PAR. The organization and operation of the swap file is controlled through the Virtual Memory option, under the Control Panel's 386 Enhanced icon.

The swap file's size determines how many applications can use the swap file. The best virtual memory performance is achieved by establishing a permanent swap file on the drive. For heavily-segmented drives or for redirected network drives, however, a temporary swap file is highly recommended.

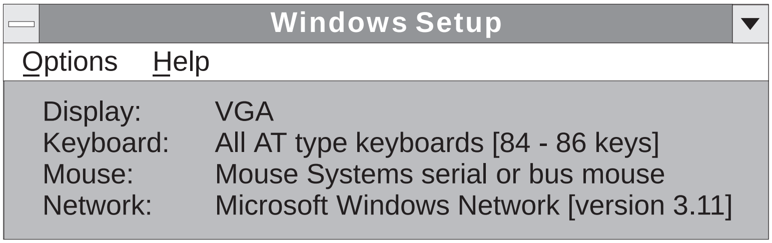
The FastDisk function also sets up a `disk cache` area in extended memory to hold data that has already been read from the hard disk. The next time the system tries to read that information, it reads the information from the system's RAM instead of accessing the hard drive. Data to be written to the disk can be temporarily stored in the cache as well. The default size of the cache is normally 512 KB. If the system has sufficient RAM (above 4 MB), the size of the cache may be extended to 1 MB.

Windows Setup

Windows Setup enables the user to add new applications after Windows has been installed. The Windows Setup icon is available through the Main group window. Double-clicking on its icon brings up the Windows Setup dialog box depicted in Figure 4.8. The window shows the current driver settings for the video, keyboard, mouse, and network functions. These settings represent the hardware that Windows detected at the time of installation.

Figure 4.8

The Windows Setup screen.



To change any of these settings, select the Options entry on the menu bar and choose the Change System Settings command. Select the system setting to be changed by clicking on the down arrow button by its entry. This action brings up a list of options available for that setting.

The Set Up Applications option is used to install new Windows applications. This option either searches the drive for an existing application, or asks the user to specify an application to be added to Windows.

The Add/Remove Windows Components option enables the user to add or remove items that are not essential to the operation of Windows. If the hard drive becomes full, these items can be removed to free up about 2 MB of memory.

Some manufacturers include a proprietary setup program for their Windows applications. In these cases, installation is as easy as choosing File from the Program Manager's menu bar, and then selecting the Run option. These actions presents a dialog box where the user can enter the path of the setup program to be installed. Whenever a new device driver is installed, or a device driver is changed, Windows must be restarted for the changes to take effect.

Other icons commonly found in the Main window include the Clipboard Viewer, the Print Manager, and the MS-DOS Prompt. The Clipboard Viewer function enables the user to cut and paste text from different Windows applications. The Print Manager controls printing for all Windows applications. The MS-DOS icon enables the user to exit from the Windows environment and execute DOS commands and programs.

Initialization (INI) Files

Objectives

When Windows 3.x is installed, several `initialization files` are copied into the Windows directory. All these files have an extension of .INI. Within these files are the default or current startup settings for various Windows components. These files are updated when changes are made, and the Save Changes Upon Exit option is set in the Program Manager's Options pull-down menu. You can also use a shortcut to update these settings by selecting the File menu from the Program Manager. From this point, press and hold the Shift key and then select Exit Windows from the File menu. This updates these files with current settings such as open groups or minimized programs.

The major initialization files are as follows:

- WIN.INI
- CONTROL.INI
- WINFILE.INI
- PROGMAN.INI
- SYSTEM.INI

There are several other INI files in the Windows directory. In fact, when a new windows application is installed it usually install its own INI file at that time as well. These files can be modified to customize, or optimize, the program's execution.

Some of the parameters in these files are modified through normal Windows menus or dialog boxes. Others can only be changed by modifying the INI file directly. The files are broken into sections that contain the individual parameters which can be altered.

Normal system functions that alter INI settings include changing Control Panel, Program Manager, or File Manager entries. Changing system settings through the Windows Setup program will also modify the contents of the INI files.

Windows provides a System Editor utility called SysEdit that can be used to modify the SYSTEM.INI, WIN.INI, CONFIG.SYS and AUTOEXEC.BAT files. This utility can be accessed by simply typing SysEdit in the Windows File/Run dialog box.

The files can be modified directly by using the DOS/Windows Text Editor, or some other word processing/text editing program. Windows has two word processing packages that can be used to modify text and INI files. These are the Notepad and Write editors, under the Accessories window. Notepad is a small ASCII editor. Write is a simple word processor package that supports document formatting, incorporation of graphics files, and object linking and embedding (OLE).

The format of all the INI files is consistent. Each INI file is divided into logical sections. Each section consists of a list of entries in the format of `keyname=value`. Each section has a name enclosed in brackets. The keyname is just a name that describes the function of the entry and is normally followed by an equals sign. It can be any combination of characters and numbers. The value entry can be any integer or string. Typical enabling entries include `On`, `True`, `Yes`, and `1`. Conversely, disabling entries are `Off`, `False`, `No`, and `0`.

```
[Section name]
keyword=value
```

Win INI

The WIN.INI file contains parameters that can be altered to change the Windows environment to suit the user's preferences. This is one of the largest INI files installed by Windows. The major sections of the WIN.INI file are Windows, Desktop, Extensions, and Colors.

The `[Windows]` Section

The `[Windows]` section can be used to alter several attributes of the environment, such as the mouse settings. Other items controlled by this section include warning beeps, printing operations, border width of windows, keyboard speed, and applications that automatically start when Windows is started.

```
[Windows]
spooler=yes
run=c:\mcafee\viruscan\vshldwin.exe
Beep=yes
NullPort=None
BorderWidth=3
CursorBlinkRate=530
DoubleClickSpeed=452
Programs=com exe bat pif
Documents=
DeviceNotSelectedTimeout=15
TransmissionRetryTimeout=45
KeyboardDelay=2
KeyboardSpeed=31
ScreenSaveActive=1
ScreenSaveTimeOut=1200
MouseThreshold1=0
MouseThreshold2=0
MouseSpeed=0
CoolSwitch=1
```

```
DosPrint=no
SwapMouseButtons=0
device=Panasonic KX-P4420,HPPCL,LPT1:
```

The `[Desktop]` Section

The `[Desktop]` section sets the appearance of the desktop and groups. It controls the appearance of the onscreen background and the position of the windows and icons.

```
[Desktop]
Pattern=0 0 84 124 124 56 146 124
Wallpaper=arcade.bmp
GridGranularity=0
IconSpacing=75
TileWallPaper=1
```

The `[Extensions]` Section

The `[Extensions]` section sets association of file types to an application. This allows icons to be used to automatically start an application when one of its documents is selected.

```
[Extensions]
cal=calendar.exe ^.cal
crd=cardfile.exe ^.crd
trm=terminal.exe ^.trm
txt=notepad.exe ^.txt
ini=notepad.exe ^.ini
pcx=pbrush.exe ^.pcx
bmp=pbrush.exe ^.bmp
wri=write.exe ^.wri
rec=recorder.exe ^.rec
hlp=winhelp.exe ^.hlp
```

The `[Intl]` Section

The international `[Intl]` section defines how different information will be formatted on the screen. It controls how dates, times, and currencies will be displayed.

```
[Intl]
sLanguage=enu
sCountry=United States
```

```
iCountry=1
iDate=0
iTime=0
iTLZero=0
iCurrency=0
iCurrDigits=2
iNegCurr=0
iLzero=1
iDigits=2
iMeasure=1
s1159=AM
s2359=PM
sCurrency=$
sThousand=,
sDecimal=.
sDate=/
sTime=:
sList=,
sShortDate=M/d/yy
sLongDate=dddd, MMMM dd, yyyy
```

The [Ports] Section

The [Ports] section is the Windows equivalent of the **DOS Mode** command. In this section, it is possible to establish and configure up to 10 logical ports.

```
 [Ports]
; A line with [filename].PRN followed by an equal sign causes
; [filename] to appear in the Control Panel's Printer Configura-
➥tion dialog
; box. A printer connected to [filename] directs its output into
➥this file.
LPT1:=
LPT2:=
LPT3:=
COM1:=9600,n,8,1,x
COM2:=9600,n,8,1,x
COM3:=9600,n,8,1,x
COM4:=9600,n,8,1,x
EPT:=
FILE:=
LPT1.DOS=
LPT2.DOS=
```

The entries at the top of this example are notes about how the system handles filenames with a .PRN extension. Windows recognizes up to three parallel ports and assigns them LPTx designations. The COMx assignments apply to the system's serial communications ports (COM1–COM4). The values to the right of the equal sign establish the speed and character parameters the port will use to conduct communications with remote devices.

The `[Fonts]` Section

Alphanumeric characters can be created in a number of different type styles. A certain set of characters may be curved in appearance while another is quite square and blocky. A given set of characters designed in a common style is called a `font`. The `[Fonts]` section describes the screen fonts that are loaded at startup. New font sets can be added to the list (from a third-party disk, for example), but they must be installed through the Fonts icon in the Control Panel.

```
 [Fonts]
Arial (TrueType)=ARIAL.FOT
Arial Bold (TrueType)=ARIALBD.FOT
Arial Bold Italic (TrueType)=ARIALBI.FOT
Arial Italic (TrueType)=ARIALI.FOT
Courier New (TrueType)=COUR.FOT
Courier New Bold (TrueType)=COURBD.FOT
Courier New Italic (TrueType)=COURI.FOT
Times New Roman (TrueType)=TIMES.FOT
Times New Roman Bold (TrueType)=TIMESBD.FOT
Times New Roman Bold Italic (TrueType)=TIMESBI.FOT
Times New Roman Italic (TrueType)=TIMESI.FOT
```

The `[FontSubstitutes]` section sets fonts (types of characters) recognized by Windows equal to another type of font. This relationship allows character sets used with other software programs to be recognized and substituted for use by an appropriate Windows font. As an example, Helvetica is a widely used character font. The Arial font in Windows is very similar to it. According to the equation in the example, if Windows encounters a document with Helvetica type in it, the Arial font will be substituted for the Helvetica characters.

```
[FontSubstitutes]
Helv=MS Sans Serif
Tms Rmn=MS Serif
Times=Times New Roman
Helvetica=Arial
```

The `[TrueType]` section defines how Windows applications will treat a special type of fonts known as True Type fonts. In this example, the system is enabled to handle true type fonts if they are available in the system.

```
[TrueType]
TTEnable=1
TTOnly=0
```

Media File Control

The `[MCI Extensions]` section defines how various media files interact with the Windows Media Control Interface (MCI). This interface is discussed in greater detail in Chapter 12, "Multimedia." In this case, a file with a .WAV extension will be routed to the Waveaudio driver.

```
 [MCI Extensions]
wav=waveaudio
mid=sequencer
rmi=sequencer
```

The `[Embedding]` Section

The `[Embedding]` section lists types of objects that can be embedded in Windows applications. The format of the entry is as follows:

```
object=description, description, program file, format
[Embedding]
SoundRec=Sound,Sound,SoundRec.exe,picture
PBrush=Paintbrush Picture,Paintbrush Picture,pbrush.exe,picture
ScreenCamMoviev2=Lotus ScreenCam Movie 2.0,Lotus ScreenCam Movie
2.0,A:\SCPLAYER.EXE,picture
MPlayer=Media Clip,Media Clip,mplayer.exe,picture
```

The [Colors] Section

The [Colors] section defines colors to use for the different Windows components. The format for defining colors is red value, green value, blue value. The range of possible values runs from 0 to 255, where 0 is minimum intensity and 255 is maximum intensity. The parts of a typical Windows window are described in Figure 4.9.

```
[Colors]
Background=0 64 64
AppWorkspace=192 192 192
Window=255 255 255
WindowText=0 0 0
Menu=192 192 192
MenuText=0 0 0
ActiveTitle=0 128 64
InactiveTitle=64 128 128
TitleText=255 255 255
ActiveBorder=0 128 64
InactiveBorder=64 128 128
WindowFrame=0 0 0
Scrollbar=192 192 192
ButtonFace=192 192 192
ButtonShadow=128 128 128
ButtonText=0 0 0
GrayText=128 128 128
Hilight=0 128 0
HilightText=255 255 255
InactiveTitleText=0 0 0
ButtonHilight=255 255 255
```

The [Windows Help] Section

The [Windows Help] section defines the placement and appearance of the Windows Help window when it is onscreen. Entries for the window occur in the following format:

- Starting x position of the window's upper-left corner
- Starting y position of the window's upper-left corner
- Default width of the window

- Default height of the window
- Window maximized (1) or minimized (0)

```
[Windows Help]
H_WindowPosition=[213,160,213,160,0]
```

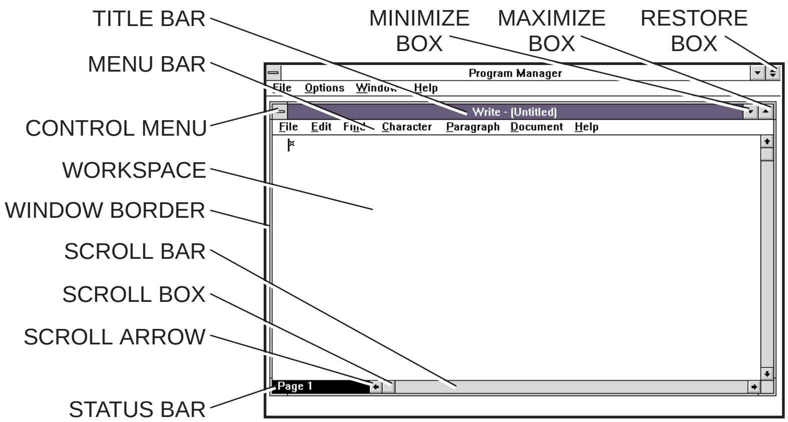


Figure 4.9

The parts of a Windows screen.

The `[Sounds]` Section

The `[Sounds]` section identifies the events that will produce sound in the system, along with the particular sound that will be generated and its description.

```
[Sounds]
SystemAsterisk=chord.wav,Asterisk
SystemHand=chord.wav,Critical Stop
SystemDefault=ding.wav,Default Beep
SystemExclamation=chord.wav,Exclamation
SystemQuestion=chord.wav,Question
SystemExit=chimes.wav,Windows Exit
SystemStart=tada.wav,Windows Start
```

Windows I/O Ports

The `[PrinterPorts]` section lists the printers, both active and inactive, that have Windows drivers installed. The entry for each printer also lists its timeout values. Timeouts are used to determine whether the printer has taken too long to respond (indicating that a problem has been encountered) and how long to wait before trying to communicate again. Both values are specified in seconds.

```
[PrinterPorts]
HP LaserJet Series II=HPPCL,LPT1:,600,1000
Panasonic KX-P4420=HPPCL,LPT1:,15,45
[devices]
HP LaserJet Series II=HPPCL,LPT1:
Panasonic KX-P4420=HPPCL,LPT1:
```

Other sections may be added to the WIN.INI file by software programs when they are installed. In the following sections, entries have been added to define the operation of the Windows Terminal communications package and Paintbrush graphics package. In the final entry, a software package has added a `[User Identification]` section to the file.

```
[Terminal]
Port=COM2
[Paintbrush]
width=640
height=480
clear=COLOR
OmitPictureFormat=0
[MS User Info]
DefName=Charles Brooks
DefCompany=Marcraft
```

SYSTEM.INI

The SYSTEM.INI file contains hardware setting information for the drivers and modules that Windows uses to configure itself when started. The key sections of the SYSTEM.INI file are the `[Boot]`, `[Keyboard]`, `[Drivers]`, and `[386enh]` sections. Setup assigns values to the entries under each of these headings, and they must be in the SYSTEM.INI file for Windows to operate properly. As a matter of fact, the SYSTEM.INI file is the only INI file that actually needs to be present in order to load Windows 3.x.

Windows Bootup Setting

The `[Boot]` section lists devices that can be changed by running the Windows Setup function. These items are used by Windows to configure itself during bootup. Values are assigned to each entry according to the selections made in the Windows Setup menu.

```
[Boot]
mouse.drv=MSCMOUSE.DRV
shell=progman.exe
network.drv=
language.dll=
sound.drv=mmsound.drv
comm.drv=comm.drv
keyboard.drv=keyboard.drv
system.drv=system.drv
386grabber=vga.3gr
oemfonts.fon=vgaoem.fon
286grabber=vgacolor.2gr
fixedfon.fon=vgafix.fon
fonts.fon=vgasys.fon
display.drv=vga.drv
drivers=mmsystem.dll WTNIPM
SCRNSAVE.EXE=C:\WINDOWS\SSMYST.SCR
```

The `[Keyboard]` section provides the startup codes for the operation of the keyboard. The keyboard files shipped with Windows are `dynamic link library` (DLL) files. These are executable modules whose code can be loaded on demand, linked at runtime, and unloaded when finished.

```
[Keyboard]
subtype=
type=4
keyboard.dll=
oemansi.bin=
```

The `[Boot.Description]` section lists the devices that can be changed through the Setup menu. There is no reason for the user to change these values directly. Doing so will disable Setup's capability to update drivers.

```
[Boot.Description]
mouse.drv=Mouse Systems serial or bus mouse
keyboard.typ=Enhanced 101 or 102 key US and Non US keyboards
network.drv=No Network Installed
language.dll=English (American)
system.drv=MS-DOS System
codepage=437
woafont.fon=English (437)
aspect=100,96,96
display.drv=VGA
```

Windows Mode Settings

The `[386Enh]` section contains information specifically related to running Windows in Enhanced mode, such as virtual memory and page swapping.

Entries in this section can appear as a filename of a virtual driver and its path, or as an asterisk (*) followed by a device name.

```
[386Enh]
keyboard=*vkd
mouse=MSCVMD.386
32BitDiskAccess=OFF
device=*int13
device=*wdctrl
network=*dosnet,*vnetbios
ebios=*ebios
woafont=dosapp.fon
display=*vddvga
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON
device=vtdapi.386
device=*vpicd
device=*vtd
device=*reboot
device=*vdmad
device=*vsd
device=*v86mmgr
device=*pageswap
device=*dosmgr
device=*vmpoll
```

```
device=*wshell
device=*BLOCKDEV
device=*PAGEFILE
device=*vfd
device=*parity
device=*biosxlat
device=*vcd
device=*vmcpd
device=*combuff
device=*cdpscsi
local=CON
FileSysChange=off
PagingFile=C:\WINDOWS\WIN386.SWP
MaxPagingFileSize=20480
COM1Irq=4
COM1Base=03F8
device=c:\dos\vfintd.386
device=vsertd.386
device=vshare.386
MinTimeslice=20
WinTimeslice=100,50
WinExclusive=0
Com1AutoAssign=2
```

The `[Standard]` section contain information that Windows uses to run in Standard mode. In this example, no values are installed.

```
[Standard]
```

The `[NonWindowsApp]` section is used to hold information required by non-Windows applications. These values affect the performance of the system when non-Windows applications are running.

```
[NonWindowsApp]
localtsrs=dosedit,ced
```

Windows Drivers

The `[MCI]` section lists the drivers that use Windows Media Control Interface. The MCI and devices that use it are described in greater detail in Chapter 12, “Multimedia.”

```
[MCI]
WaveAudio=mciwave.drv
Sequencer=mciseq.drv
CDAudio=mcicda.drv
```

The `[Drivers]` section contains names assigned to the installable driver programs. These files control and communicate with the system's hardware options.

```
[Drivers]
timer=timer.drv
midimapper=midimap.drv
WTNIPM=wtnipm.drv
```

CONTROL.INI

The CONTROL.INI file contains settings for the Control Panel. This file is relatively small, and mostly contains current color settings and screen saver settings. The major sections of the CONTROL.INI file are the Current, Color Schemes, and Custom Colors.

Windows Colors and Patterns

The `[Current]` heading parameter defines the current color scheme name.

```
[Current]
color schemes=Emerald City
```

The `[Color Schemes]` parameters define color values for each Windows component. Each scheme is identified with a name and carries a set of default color values for the various screen components. The following example shows two sample schemes, Arizona and Emerald City. Even the minimum installation, however, will have several default schemes and any optional schemes that have been defined:

```
[Color Schemes]
Arizona=804000,FFFFFF,FFFFFF,0,FFFFFF,0,808040,C0C0C0,FFFFFF,4080FF,C0C0C0,
➥0,C0C0C0,C0C0C0,808080,0,808080,808000,FFFFFF,0,FFFFFF
Emerald
```

```
City=404000,C0C0C0,FFFFFF,0,C0C0C0,0,408000,808040,FFFFFF,408000,808040,
➥0,C0C0C0,C0C0C0,808080,0,808080,8000,FFFFFF,0,FFFFFF
```

The [Custom Colors] parameters specify colors that have been added to the basic color palette. This example shows a single custom color has been defined (B–D and beyond are default values). These values are adjusted through the Control Panel's Color icon.

```
[Custom Colors]
ColorA=968AFF
ColorB=FFFFFF
ColorC=FFFFFF
ColorD=FFFFFFcx
```

The patterns for different bit-mapped colors are specified in the [Patterns] section. These values are set through the Desktop icon.

```
[Patterns]
(None)=(None)
Boxes=127 65 65 65 65 65 127 0
Paisley=2 7 7 2 32 80 80 32
Weave=136 84 34 69 136 21 34 81
Waffle=0 0 0 0 128 128 128 240
Tulip=0 0 84 124 124 56 146 124
```

The [Installed] Section

The currently installed version of Windows and its installed printers are listed in the [Installed] section.

```
[Installed]
3.1=yes
PSCRIPT.DRV=yes
HPIII522.WPD=yes
PSCRIPT.HLP=yes
TESTPS.TXT=yes
HPPCL.DRV=yes
UNIDRV.DLL=yes
FINSTALL.DLL=yes
FINSTALL.HLP=yes
UNIDRV.HLP=yes
```

Multimedia Settings

Values associated with items under the Control Panel's Multimedia icon are specified in the `[MMCPL]` section.

```
[MMCPL]
NumApps=13
X=0
Y=0
W=430
H=240
ODBC=C:\WINDOWS\SYSTEM\ODBCINST.DLL
```

Screen Savers

The `[Screensaver.*]` section defines how the selected screen saver is to be displayed, including speed and density of the objects moving on the screen. The Screen Saver feature is provided to prevent static displays from remaining on the system's monitor for too long. Over time, the display will become permanently etched in the display's screen and ruin it.

```
 [Screensaver.Stars]
Density=25
WarpSpeed=4
PWProtected=0
```

The `[Drivers.Desc]` Section

The information in the `[Drivers.Desc]` section describes the functions of the Windows Midimapper and Timer.

```
[Drivers.Desc]
mciwave.drv=[MCI] Sound
mciseq.drv=[MCI] MIDI Sequencer
timer.drv=Timer
midimap.drv=MIDI Mapper
```

WINFILE.INI

The WINFILE.INI contains settings and defaults for the File Manager. This file essentially contains only one section, Settings. Its parameters specify default settings that are used during File Manager operations.

These deal with confirmations, which when enabled, are required before the requested operation is performed. When Windows is installed, all these confirmations are enabled. There is also a parameter that indicates whether the filenames are displayed in upper or lowercase. Windows defaults to lowercase.

```
[Settings]
Window=0,0,640,480, , ,1
dir1=0,0,500,279,-1,-1,1,30,201,1814,250,C:\IC100\CHAP3\*.*
```

In this example, the `Windows=` setting describes the size of the window on the screen and that it is maximized when opened. The `dir1=` line indicates the current directory settings.

PROGMAN.INI

The PROGMAN.INI file contains settings for the Program Manager. It can have three sections: Settings, Groups, and Restrictions.

The `[Settings]` Section

The `[Settings]` section holds general configuration information. This example depicts typical entries in this section. The first line indicates the position of the window onscreen, with the final number indicating whether the window is maximized.

```
[Settings]
Window=-4 61 641 467 1
display.drv=vga.drv
Order= 2 3 1 7 22 4 5 15 19 8
AutoArrange=0
SaveSettings=1
```

The `[Groups]` Section

The `[Groups]` section defines the valid group names, sizes, and locations for the group icons. A group (GRP) file contains information about the objects included in the program group. This section also indicates the order in which the group icons are presented onscreen.

```
[Groups]
Group1=C:\WINDOWS\MAIN.GRP
Group2=C:\WINDOWS\ACCESSOR.GRP
```

```
Group3=C:\WINDOWS\GAMES.GRP
Group4=C:\WINDOWS\STARTUP.GRP
Group5=C:\WINDOWS\APPLICAT.GRP
Group7=C:\WINDOWS\WPW51US.GRP
Group8=C:\WINDOWS\CORELVEN.GRP
Group15=C:\WINDOWS\MCAFEE.GRP
Group19=C:\WINDOWS\MCAFEEVI.GRP
Group22=C:\WINDOWS\MARCRAFT.GRP
```

The `[Restrictions]` Section

The `[Restrictions]` section is optional, and may not be included in all PROGMAN.INI files. This section can be added, however, to prevent access to certain Program Manager functions and menus. The parameters in this section override any other configuration settings.

```
[Restrictions]
NoRun=1
NoClose=1
NoSaveSettings=1
NoFileMenu=1
EditLevel=4
```

The `NoRun=1` and `NoClose=1` lines disable the Run and Close options in the File menu. Only icons can be used to run programs. Similarly, the `NoFileMenu=1` command disables the File menu altogether.

The `NoSaveSettings=1` command disables the Save Settings entry in the Options menu. The `EditLevel=` command provides five editing-level settings that restricts user access to the Program Manager. The 0 level enables the user free access to all options. The level 4 restriction prevents users from changing any program item information.

DOS and Windows

In many instances, it is desirable to run a DOS application from the Windows environment. From the earlier discussion about how Windows, DOS, and the BIOS interact, it should be apparent that running a DOS program from this position can be difficult. DOS applications can be run from inside a Windows window, or from a full-screen mode. However, these applications can only be operated within a window while in 386 Enhanced mode. The About Program Manager option, from the Program Manager's Help menu, can be consulted to determine which mode Windows is currently operating in.

When a non-Windows application runs in 386 Enhanced mode, Windows establishes a virtual 86 environment for it that simulates 8086 real-mode operation. Within the virtual 86 machine, Windows establishes a DOS environment with the drivers that were in operation when the system left DOS and entered Windows. Windows keeps a picture of the original DOS environment stored in memory. Under this organizational structure, the Windows Kernel and Windows applications run in protected mode and remain active while each non-Windows application runs in a virtual 86 timeslice. This requires Windows to shift between protected mode for Windows applications and virtual 86 mode for DOS applications.

Under Windows 3.x, the operating system gives control to the Windows application when it is running. When the application completes its task, it returns control of the system back to the operating system. The operating system then gives control to the next scheduled application. This transfer of control from the operating system to the application during multitasking operations is called `cooperative multitasking`. The various tasks cooperate with each other, more or less, in sharing control of the system.

In the case of DOS applications running in Windows, the operating system does not give total control of the system over to the application. Instead, Windows runs the application in a timeslice and then moves on to the next application at the end of the designated time period. This type of multitasking where the operating system retains control is referred to as `preemptive multitasking`. Even though the system runs the DOS session in a preemptive mode, the Windows applications running on the system operate in a cooperative mode.

Running DOS programs from inside a window is more resource intensive than running them from a full screen. The application can be switched to a full-screen operation by pressing the Alt and

Enter keys simultaneously. Windows can be configured to run multitasking operations in the foreground, while DOS programs are operating in the background. Because of the interactions between DOS and Windows, programs that directly access the hard drive, and by-pass Windows safeguards, should be avoided in Windows-based operations.

Windows for Workgroups 3.11

The 3.11 version of Windows for Workgroups (WfW) adds networking capabilities to the Windows environment. The updated File and Print Managers allow resources such as printers and directories to be shared. When directories are enabled for sharing, other terminals on the network can have access to your files. This is particularly useful when you are working in a group setting, where various members of your team or workgroup need access to things you are working on.

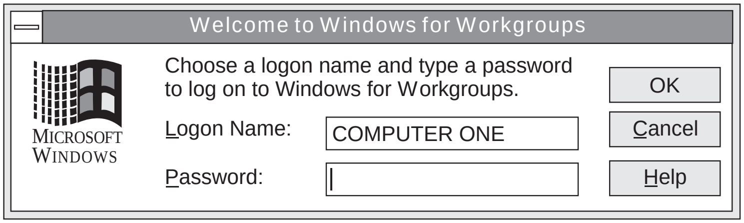
Security

When a computer is networked to another computer, a certain amount of security is lost. Because a physical path has been established into the computer, data stored in the system is now potentially available to anyone on the network. Therefore all networking software includes some type of data security system. In Windows 3.11, the data on a particular computer is protected by a logon password. When Windows is started, a Logon dialog box, like the one in Figure 4.10, pops up asking for a password to get on to the system. This password can be up to 14 characters in length. As with any security system, the password should be easy to remember, and it should be changed periodically.

After the user has logged on, he may connect to shared directories and resources around the network. Some resources may require an additional password to be accessed. In Windows, it is possible to create a password list under a logon name that will allow the user to connect to these resources without entering the password—that is, after you have access to a resource you can always have access to it as long as its password is not changed).

Figure 4.10

The Windows 3.11 Logon dialog box.



The logon process also allows the system to create a custom environment for each user that may work at any given station. User-specific information can be stored in the system and recalled when that person logs on.

File Manager

The 3.11 File Manager window contains four new icons dedicated to network management. These icons are located on a toolbar near the top of the screen. The new icons are the Connect Network Drive, Disconnect Network Drive, Share As, and Stop Sharing, illustrated in Figure 4.11. Only shared directories and resources can be accessed across the network. The Share As icon notifies other potential users that this resource or directory can be accessed.

To access a shared resource, the Connect Network Drive icon must be activated. When engaged, this icon creates a new logical drive on the local machine to handle the shared directory. File Manager will assign the directory to the next logical drive available in the local system. The path to the shared directory must contain a little more information than the path to a local directory. The remote path must include the remote computer's name and shared directory name. The format of a shared path is `\\computer name\directory name`. The Disconnect Network Drive icon is used to quit the connection to a shared resource. Likewise, the Stop Sharing icon is used to discontinue access to a locally shared resource or directory.

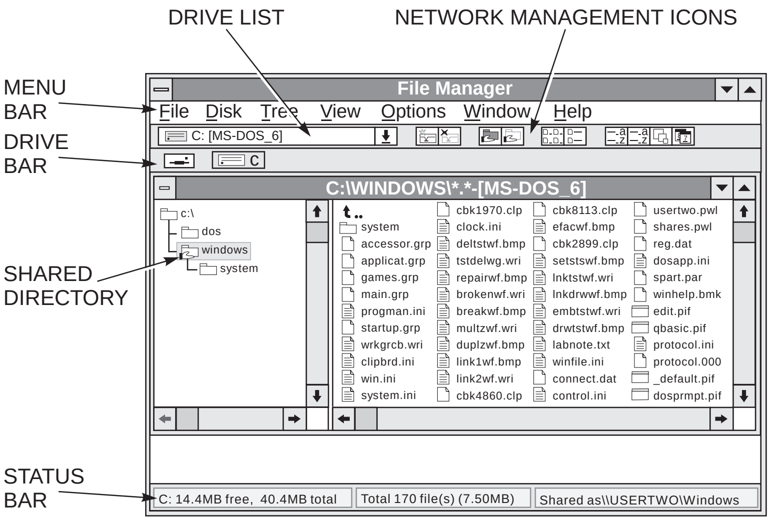


Figure 4.11

The Windows for Workgroups File Manager window.

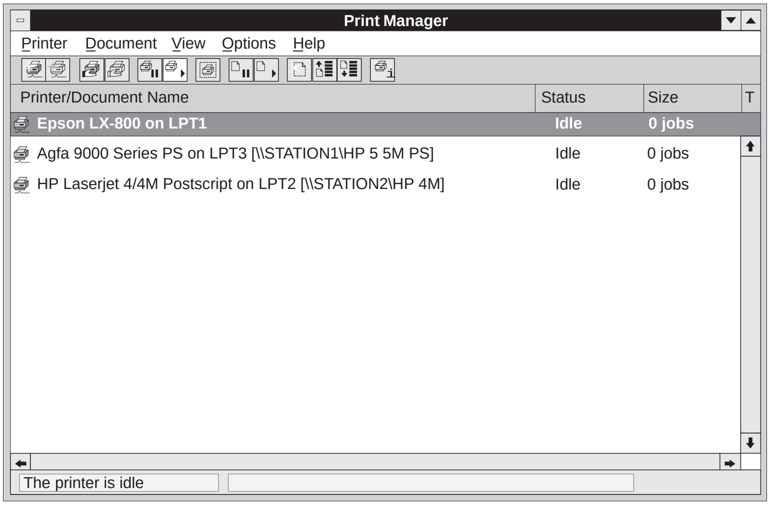
Print Manager

Like the 3.11 File Manager, the Print Manager window has been modified to accommodate networking. In particular, two new icons have been added to the Print Manager toolbar: Share Printer As, and Stop Sharing Printer (see Figure 4.12). Clicking on the printer sharing icons allows the local system to share its printer with other stations on the network, and allows the local system to have access to other printers and printer types around the network. The local printer may be a small, ink-jet or personal laser printer, for example, and the more powerful network printers used for special projects or heavy output may be available at remote locations.

The Share Printer As icon is clicked to grant remote units access to the local printer. The icon is also clicked to update the password for using the printer. The local printer is connected to a serial or parallel port in the back of the local computer, and is identified by the proper DOS handle (COM1, LPT1, and so forth). The remote printers are connected to other units around the network, and are logically linked to a port name in the local computer. When the port name is active in Print Manager, no local printer hardware is required. The network redirects the printer operation to the remote printer's port. Like the File Manager paths, network printer ports require a network path in the format of `\\Computer Name\Share Name`.

Figure 4.12

The Windows for Workgroups Print Manager window.

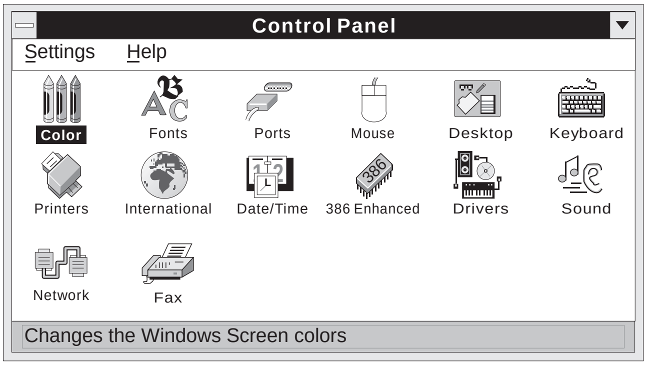


Control Panel

Two new icons have been added to the Control Panel window, as illustrated in Figure 4.13. These icons are the Network and the Fax options.

Figure 4.13

The Windows 3.11 Control Panel.



The Network option is used to establish and change the local computer's network name, logon password, and workgroup that the local system is operating in. This option also enables you to specify how the local CPU's time is allocated between local duties and resource-sharing operations. Figure 4.14 shows the WfW Network Settings dialog box.

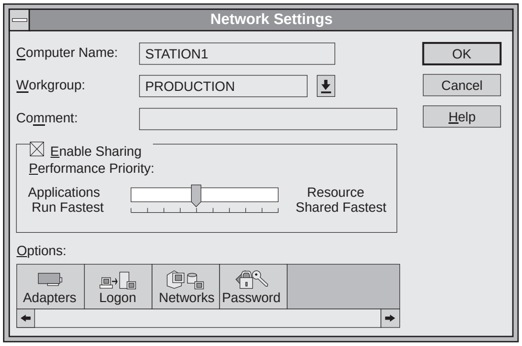


Figure 4.14

The Network window.

Double-clicking on the Fax icon opens the Fax Modems window, shown in Figure 4.15. This window enables the user to configure the system so that it can be used to send and receive faxes. The icon can be used to set up a local fax/modem or a shared fax/modem located remotely on the network. To share a fax modem on the network, a special directory must be established to store outgoing faxes. The directory must be shared and have full access privileges to be used with the remote fax/modem.

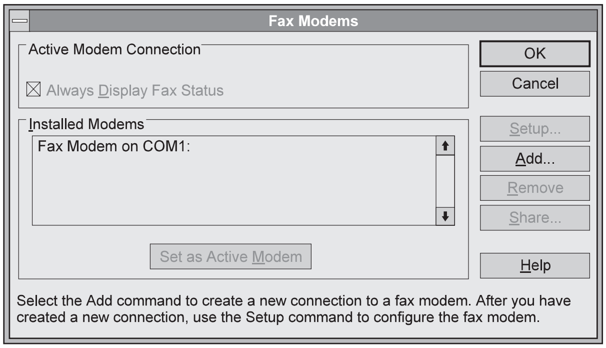


Figure 4.15

The Fax Modems window.

Windows 95

In 1995, Microsoft released a radically different-looking Windows environment called Windows 95. This Windows featured many new and improved features over previous versions. Win95 offered improved multimedia support for video and sound file applications, Plug and Play hardware support, 32-bit advanced multitasking functions, improved email and fax capabilities through Microsoft Exchange, and the Microsoft Network for easy wide area network (WAN) usage.

Even though Windows 95 is optimized for running 32-bit applications, it is still fully compatible with 16-bit Windows 3.x and DOS applications. The only real concern when installing Windows 95 over either of these operating systems is that the system has the hardware resources needed to run Windows 95.

Windows 95 offers full built-in Plug and Play (PnP) capability. When Win95 is combined with a hardware system that implements PnP BIOS, expansion slots, adapter support, and is supported with PnP adapter drivers, fully-automated configuration and reconfiguration can take place.

Win95 also does away with the 8+3 character filename system implemented under DOS. In Windows 95, long filenames of up to 255 characters can be used; therefore they can be more descriptive in nature. When these filenames are displayed in non-Windows 95 systems, they are truncated (shortened) and identified by a tilde character (~). This mark shows that the filename is being displayed in a shortened manner. Customers with older operating systems may overlook files because they are saved in this manner. Win95 also uses the right mouse button to pop-up a menu of functions, such as arranging icons, creating short cuts, and changing display properties.

Installing Windows 95

Windows 95 must be installed over an existing operating system, such as MS-DOS or Windows 3.x. In particular, the Windows 95 installation program must find a recognizable MS-DOS FAT-16 partition on the drive. This prevents it from being installed over some other operating system. The system must be at least an 80386DX or higher machine, operating with at least 4 MB of RAM (8 MB is recommended). The system should also possess a mouse and a VGA or better monitor. The system's hard drive should have at least 20 MB of free space available to successfully install Windows 95.

With the Windows 95 Setup disk, or CD-ROM in the drive, the Windows 95 Setup routine can be executed from the DOS command line, from the Windows 3.x Program Manager's Run box, or from the File Manager window. The preferred method is to run the Setup program from Windows 3.x.

To run the Setup program from Windows 3.x, follow these steps:

1. Boot the computer and start Windows.
2. Insert the Windows 95 Start Disk (Disk 1) in the A: drive, or place the Windows 95 CD in the CD-ROM drive.
3. Open the File Manager and select the proper drive.
4. Double-click on the SETUP.EXE file entry.
5. Follow the directions from the screen and enter the information requested by the program for the type of installation being performed.

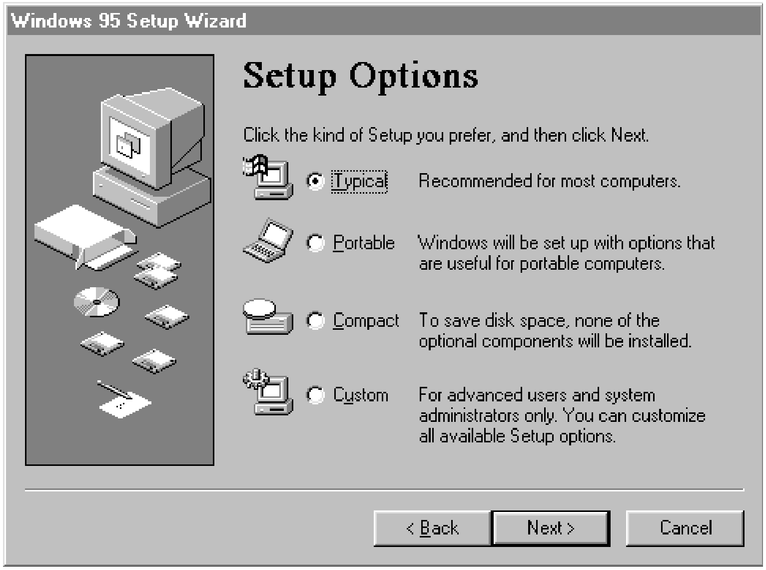
To run the Setup program from DOS, follow these steps:

1. Boot the computer.
2. Insert the Windows 95 Start Disk (Disk 1) in the A: drive, or place the Windows 95 CD in the CD-ROM drive.
3. Move to the drive that contains the Windows 95 Installation files.
4. At the DOS prompt, type **Setup** and press Enter.
5. Follow the directions from the screen and enter the information requested by the program for the type of installation being performed.

The Setup program provides options for performing Typical (default), Portable, Compact, and Custom installations. Figure 4.16 depicts the Windows 95 Setup Wizard's Setup Options screen. The Typical process installs most of the Windows 95 files without intervention from the user. The Portable option installs those options most closely associated with portable computer systems. The Compact option is a minimal installation for those units with limited disk space available. The Custom option enables the user to make customized selections for most device configurations.

Figure 4.16

The Windows 95 Setup Options Screen.



One custom installation feature includes retaining the Windows 3.x Program Manager for those who are comfortable with the look and feel of the 3.x environment. This is accomplished by selecting the Custom setup option, moving into the computer Settings dialog box and clicking on the User Interface option. At this point, click Change and select Windows3.1. The Program Manager can be run by setting up a shortcut to the PROGMAN.EXE file. However, Windows 95 folders and some other desktop-related functions will not work with Program Manager. Also, it will now be necessary to Shut Down Windows 95 after leaving the Program Manager.

If Setup detects the presence of a Windows 3.x operating system, it asks whether it should install its files in the same directory. If the prompt is answered with Yes, Windows 95 will act as an upgrade over the existing Windows structure. It will obtain existing configuration information from the SYSTEM.INI, WIN.INI, and PROTOCOL.INI files and move it into a new configuration are known as the Registry. This will allow these settings to work automatically when Windows 95 is first started.

Windows 95 also migrates the contents of the existing Windows 3.x Group (.GRP) files into the Registry during installation. Since Windows 95 rummages around in these files, both the .INI and .GRP files from the original Windows 3.x setup should be backed up before installing Windows 95.

Because Windows 95 draws from the existing DOS or Windows 3.x structure when it is being installed, an interruption, or a crash during the installation process might leave the system with no workable operating system in place. If this occurs, it will be necessary to boot the system from a bootable floppy disk and reinstall Windows 95 from that point. If the MS-DOS version that Setup detects on the hard drive is older then 3.1, it will present an “Incorrect MS-DOS version” message on the screen. If so, the hard drive will need to be reformatted with a more current version of DOS.

If the answer to the installation directory question is No, you are asked where the Windows 95 files should be installed. By installing Win95 in a new directory, it will be possible to preserve the old DOS or Windows environment. To boot to both operating systems, it is necessary to configure the system with dual boot options.

Dual-Boot Configuration

By establishing a `dual-boot configuration`, it is possible to install Windows 95 on an existing system and still retain the original operating systems. As previously stated, the first step in establishing a dual-boot system is to install the copy of Windows 95 into a new directory.

To dual boot with DOS, the system must have a copy of DOS 5.0 or higher running. In addition, the setting for the Windows 95 MSDOS.SYS file’s `BootMulti=` entry must be set to a value of `1`. This can be done by bringing the file into a text editor, such as Notepad, and change the setting to the desired value. After rebooting the system, it will be possible to boot into the old DOS/Windows 3.x environment by pressing the F4 function key when the Starting Windows message appears during bootup.

If Windows 95 is already installed in the system, it will be necessary to copy the IO.SYS and MSDOS.SYS files to a floppy, rename them IO.DOS and MSDOS.DOS, and then copy them back into the root directory. They will also need to be handled as any other hidden, read-only, system files (use the `Attribute command` to read and copy them). It will be necessary to perform the same copy/rename/copy operations on the existing COMMAND.COM file. Finally, create new CONFIG.DOS and AUTOEXEC.DOS files that are appropriate for the version of DOS you are using in the system. Just restart the system to run DOS or Windows 95. The reasons for changing the filenames will become more apparent as the chapter explains the organizational structure that Windows 95 takes.

Win95 Startup

Objectives

Unlike previous versions of Windows, Win95 does not overlay a DOS structure. Instead, Win95 takes over the DOS bootup functions as a normal part of its operation. This seamless bootup may be convenient, but can offer some interesting problems when the system will not boot up—there's no DOS level to fall back to for troubleshooting purposes.

Basically, the Win95 portion of the bootup sequence occurs in five phases:

Phase 1 - Bootstrap with the BIOS

Phase 2 - Loading DOS drivers and TSR files

Phase 3 - Real-mode initialization of static virtual device drivers (VxDs)

Phase 4 - Protected-mode switch over

Phase 5 - Loading of any remaining VxDs

VxDs are protected-mode drivers that allow multiple applications to access a system hardware or software resource. The `x` in the abbreviation represents a particular type of driver—that is, VDD is a display driver, VPD is a printer driver, and so forth).

There are two types of VxDs; those that must be `statically loaded` and those that may be `dynamically loaded`. Static VxDs are loaded into memory and stay there while the system is operating. All virtual device drivers are loaded this way in Windows 3.x. The problem is that they use a lot of memory. Win95 dynamically loads some drivers into memory while they are needed. Win95 VxDs files have an extension of .VXD; Win 3.x drivers retain the .386 extension.

Phase 1: The Bootstrap Process

During the bootstrap process, the BIOS is in control of the system and functions as described in the preceding chapter. However, most newer BIOS contain Plug and Play features designed to work with the Windows 95 architecture. In particular, the BIOS passes configuration information about the system to the Win95 Configuration Manager. At startup, the PnP BIOS checks the CMOS RAM to determine which PnP devices should be activated and where their PnP should be stored, as well as what their DMA, INT, and I/O addressing assignments should be. Upon completion of the configuration check operation, the BIOS configures the PnP cards and the intelligent system board devices. The BIOS then performs the POST and Initialization functions for the system.

Phase 2: Loading DOS Drivers and TSR Files

After the disk boot, Win95 checks the system's hardware profile to determine its actual configuration. This profile is a function of the detection process of the BIOS during the initialization phase. At this point, the CONFIG.SYS and AUTOEXEC.BAT files are executed. These files are present to maintain compatibility with applications written for earlier operating systems and environments. In the Windows 95 environment, the CONFIG.SYS file contains a name listing for each hardware profile used.

In the next phase, Win95 loads the WIN.COM file to control the loading and testing of the Win95 core components.

Phase 3: Initializing Static VxDs

This is followed by loading the VMM32.VXD Virtual Machine Manager and, finally, the SYSTEM.INI file. SYSTEM.INI is loaded so that its information can be used to maintain compatibility with nondynamic VxDs

The VMM32.VXD file creates the virtual environment and loads the VxD files. It contains a list of all the VxD files the system requires. These files are stored in a branch of the Windows 95 Registry. The Registry is a centralized, improved replacement for the old Windows 3.x INI files. If the value in the listing is represented by a `StaticVxD=` statement, the VMM32.VXD file loads and initializes it in real mode. It also statically loads any VxDs that have a `device=xxxVxD` entry. The dynamic VxD files are not loaded by the VMM32.VXD file.

Phase 4: Protected-Mode Switch Over

After loading all the static VxDs, the VMM32.VXD file shifts the microprocessor into protected-mode operation and begins loading the protected-mode components of the system.

The Configuration Manager is loaded and initialized with configuration information from the PnP BIOS. If no PnP information is available, the Configuration Manager develops a PnP tree by loading dynamically loadable drivers. After the tree is in place, the Configuration Manager reconciles the configuration information for each device, resolves any conflicts, and then reconfigures any devices necessary.

Phase 5: Loading Remaining Components

Following the initialization process, the final Win95 components are loaded into the system. These components include the following:

- The KERNEL32.DLL and KERNEL386.EXE files
- The GDI.EXE and GDI32.EXE files
- The USER.EXE and USER32.EXE files

- All font and other associated resources
- The WINI.INI file
- The Win95 shell and desktop files

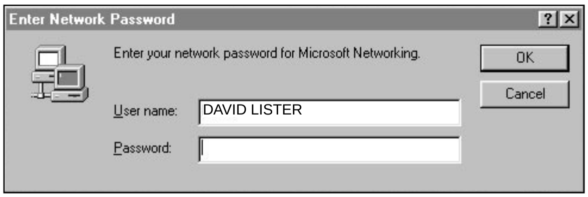
The KERNEL32.DLL and KERNEL386.EXE files contain the Win95 core and load its device drivers. The GDI files provide the base of the graphical device interface; the USER files provide the user interface. The GDI files graphically represent and manage the system hardware devices. The WIN.INI file is included for compatibility functions with older software. These files are retained for use with older 16-bit applications and are not necessary for the operation of Windows 95 applications.

When the shell and desktop components are loaded, the system displays an onscreen prompt for the user to log on, as shown in Figure 4.17. Similar to the logon process associated with networked systems, the Win95 logon allows the operating system to configure itself for specific users. Normal logon involves entering a username and password. If no logon information is entered, default values are loaded into the system. The Logon dialog box only appears if the system is in use with a network, or when there are settings that the user can customize.

Windows 95 possesses system startup files that replace the DOS files described in the preceding chapter. The Win95 version of IO.SYS is a real-mode operating system that replaces the DOS version. It also takes over many of the functions associated with the CONFIG.SYS file. An MSDOS.SYS file is created to retain compatibility. However, the VMM32 and VxDs take over control of the system from the IO.SYS file during the startup process.

Figure 4.17

The Windows 95 Logon dialog box.



Although the CONFIG.SYS and AUTOEXEC.BAT files are not required to startup Windows 95, they are retained from the previous compatibility with older applications. Entries in the CONFIG.SYS file override the values in the Win95 IO.SYS file, however. The IO.SYS file also handles some of the AUTOEXEC.BAT commands. In both cases, the system uses REM statements to deactivate those CONFIG.SYS and AUTOEXEC.BAT functions that are implemented in the IO.SYS file. Similarly, the functions of the SYSTEM.INI and WIN.INI files have been moved to the Windows 95 Registry. As with Windows 3.x, the SYSTEM.INI, WIN.INI, PROTOCOL.INI, CONFIG.SYS, and AUTOEXEC.BAT files can be modified through the *System Editor* in Windows 95. The *SysEdit* utility can be accessed by selecting the SysEdit option in the Start/Run dialog box.

Win95 Desktop

When Windows 95 is started, it produces the basic desktop depicted in Figure 4.18. The new user interface features four basic icons: My Computer, Network Neighborhood, Recycle Bin, and the Start button.

The My Computer icon enables the user to see the local system's contents and manage its files. The Network Neighborhood icon provides information about the world around the system, when it's used in a networked environment. The Recycle Bin is a storage area for deleted files, which enables you to retrieve such files if they are deleted by mistake. This icon should always be present on the Desktop. It can only be removed through the Registry. If it icon is missing, there are two alternatives to restoring it: establish a short cut to the Recycle Bin by using a new icon; or reinstall Windows 95. This action will always place the Recycle Bin on the desktop. The Start button is used to accomplish several different tasks depending on the context of the operation. The Start button, for example, is used to start programs, alter system settings, and open documents.

All operations begin from the Start button. When you click on the button, a pop-up menu of options appears (see Figure 4.19). This menu normally contains the options Programs, Documents, Settings, Find, Help, Run, and Shut Down.

Placing the cursor over designated menu items will cause any submenus associated with that option to pop up onscreen. An arrow head to the right of the option indicates a submenu is available. To open the selected item, just click on it; its window will appear onscreen.

Figure 4.18

The Windows 95 desktop.

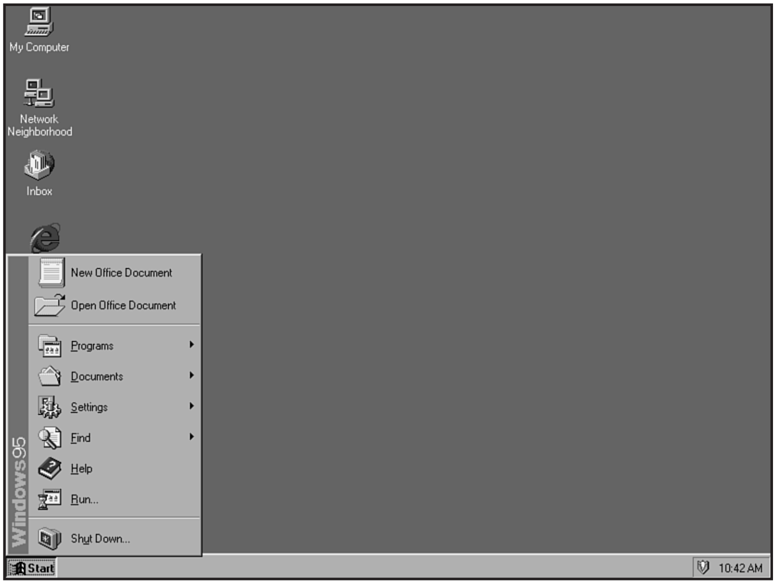


Figure 4.19

The Start button menu.

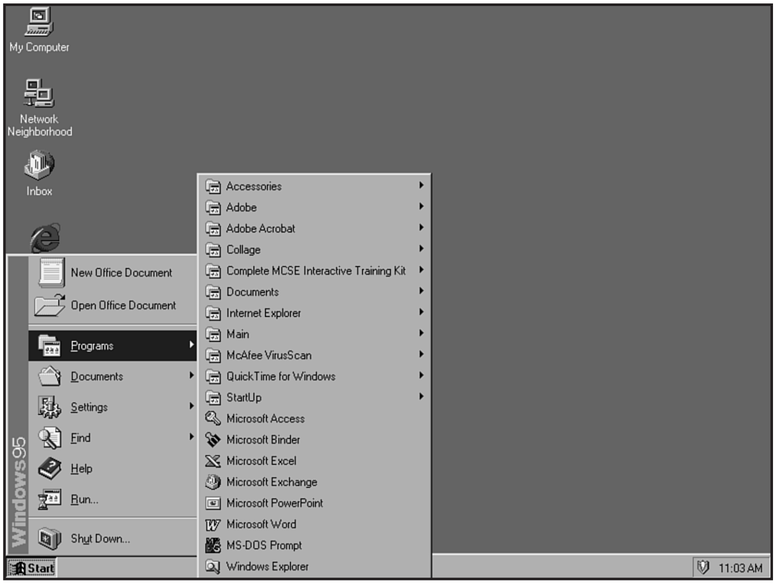
Just to the right of the Start button is an area called the taskbar. This area is used to display all the applications currently open.

Each time a program is started, or a window is opened, a corresponding button appears on the taskbar. To switch between applications, just click on the desired program button to make it the active window.

The Programs submenu, depicted in Figure 4.20, has several options that include Accessories, On-line Services, Start Up, Windows 95 Training, MS-DOS Prompt, and Windows Explorer.

Figure 4.20

The Programs submenu.



Several functions from Windows 3.1x are combined under the Programs option. The Windows groups appear when this option is selected. The File Manger function from 3.1x is performed by the Windows Explorer option. By clicking on it, the system's directory structure will appear, as shown in Figure 4.21.

Like the Windows 3.x File Manager, the Windows Explorer enables the user to copy, move, and erase files on any of the system's drives. Its screen is divided into two parts. The left side displays the directory tree showing all the directories and subdirectories of the system's available drives. In Windows 95, directories and subdirectories are referred to and depicted as folders (and subfolders). Any drive or directory can be selected by moving the cursor so that it is over the desired icon or folder and then clicking on that icon or folder. The contents of the folder can be expanded by clicking on the plus sign beside the folder. Conversely, the same folder can be contracted by clicking on the minus sign in the same box.

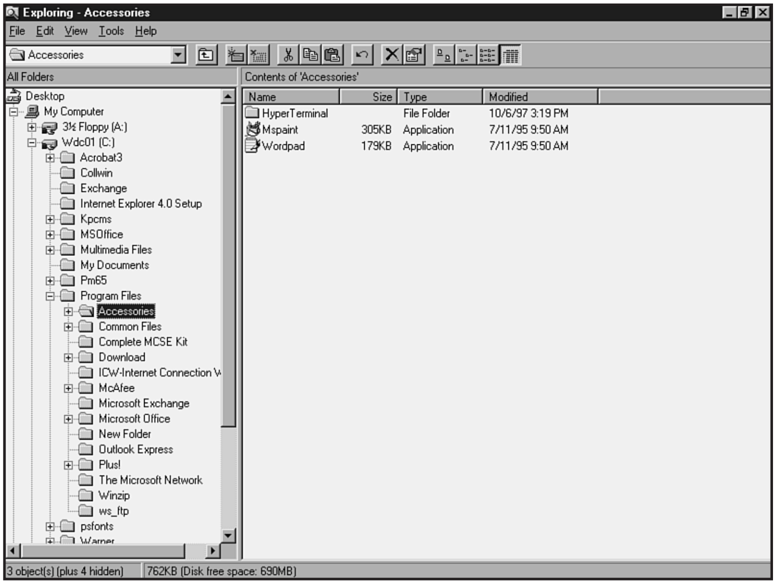


Figure 4.21

The system's directory structure.

The right side of the Windows Explorer screen displays the files of the selected directory. The status bar at the bottom of the screen provides information about the number and size of the files in the selected directory.

It is possible to display multiple directories on the Explorer screen. This feature makes it easy to perform file operations by just opening another window. Like Windows 3.x, Windows 95 provides drag-and-drop file copies and moves for single and multiple files. The Windows Explorer is also used to perform other DOS-like function such as formatting and copying disks.

The MS-DOS Prompt is also accessed through the Programs option.

The Start Menu's Documents entry displays a list of documents previously opened.

The Settings option, from the Start button menu, displays values for the system's configurable components. It also combines previous Windows functions. The Control Panel and Print Manager functions can be found here, as well as access to the Win95 taskbar. The Win95 taskbar uses new button icons for the Minimize, Maximize, and Close functions, as described in Figure 4.22. Each time a program is started, a button representing that program is displayed on the lower line of the taskbar. Changing from one program to another is as simple as clicking on the button of the program you want. The button will disappear from the taskbar if the program is closed.

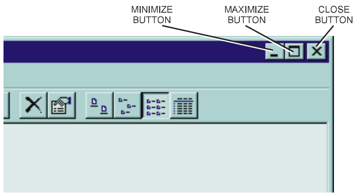


Figure 4.22

Windows 95 Minimize, Maximize and Close functions.

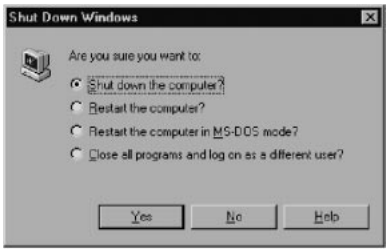
The Find utility is used to locate folders, files, mail messages, and shared computers. The Find function can be accessed directly from the Start menu on the taskbar, or it can be reached by right-clicking the My computer icon. The selection from the Start menu allows files, folders, and computers to be searched for. the My computer option searches only for files and folders. To locate a file, type its name in the Named window, tell the system which drive or drive to look in, and click Find. Standard DOS wildcards can be included in the search name.

The Run option is used to start programs or open folders from a command line. The *Help* file system provides information about many Win 95 functions and operations. It also supplies an exhaustive list of guided troubleshooting routines for typical system components and peripherals.

The Start button is also used to correctly shut down Windows 95. The Shut Down option from the Start menu shuts down the system, restarts the computer, or logs the user off. It must be used to avoid damaging files and to ensure that your work is properly saved. When it is clicked, the Shut Down Windows dialog box depicted in Figure 4.23 appears. After selecting an option from the dialog box, the unit tells you to wait until you receive a screen message telling you that it is okay to turn the system off.

Figure 4.23

The Shut Down dialog box.



Windows 95 Control Panel

The Control Panel in Windows 95 is located under the My Computer icon. The Control Panel window, shown in Figure 4.24, contains icons for every device attached to he system. Each icon accesses the configuration information for the system's installed devices specific to its type. The icons are visually different from those used in the Win3.x Control Panel.

Figure 4.24

The Windows 95 Control Panel.

One of the main functions of the Control Panel is to enable users to customize the Win95 desktop. This customization includes setting screen colors, changing the Windows wallpaper, and selecting screen savers. Wallpaper is the pattern that shows behind the various application windows. Screen savers are screen displays that remain in motion while the system is setting idle. This utility prevents a single display from remaining on the screen for a prolonged time. This keeps the image from being "burned into" the screen. When this happens, the image becomes a permanent ghost on the screen and the monitor is ruined.

The Control Panel is also used to assign ports for printers and mice, as well as specifying how various peripheral devices respond. The Add New Hardware and Add/Remove Programs icons are used to set up interrupt and port assignments for new hardware devices, and to install device driver programs to support the hardware.

Installing Hardware and Software

Windows 95 is designed to try to help the user set up new hardware components that may be added to the system. An icon named `Add New Hardware` can be found under the Control Panel option of the Settings menu. Double-clicking on this icon activates the Win95 Wizard, depicted in Figure 4.25. The wizard program is designed to guide you through hardware setup steps. The new card or device should already be installed in the system before running this procedure.

Like the Hardware Wizard, Win95 offers the user assistance in installing new programs. The Add/Remove New Programs icon under the Control Panel option is used to install new programs automatically.

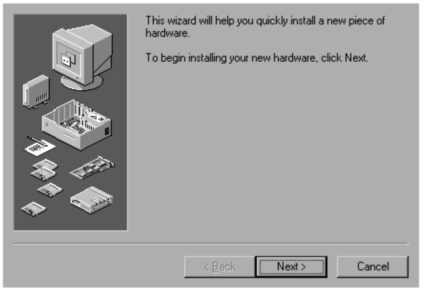


Figure 4.25

The Win95 Wizard.

Setup

The Windows Setup function is located under the Control Panel's Add/Remove Programs icon. Configuration changes can be made through its dialog boxes. Some component manufacturers include a proprietary setup program for their Windows 95 applications. The Apply option searches the available drives for applications and installs them in the system.

DOS and Windows 95

DOS-based applications are installed in Windows 95 by simply running their executable file from the Run dialog box or Windows 95 Explorer. If the file has never been run under Windows 95, the operating system creates a default entry in its APPS.INF file for that program. A copy of the new entry is also used to create a .PIF file for the application.

Once the APPS.INF entry has been created, it can be accessed and modified through the Properties window for that application. These properties windows replace the PIF editor used in previous versions of Windows. The Properties window contains six tabs that allow the operation of the application to be modified. These tabs are:

- General
- Program
- Font
- Memory
- Screen
- Misc

The Program tab allows the user to define where the DOS program is located, what it is called, and how it should be displayed. The tab's Run entry is used to establish the initial window size setting for the application. Options for this setting include normal window, maximized window, and minimized window.

Nearly every DOS-based program should run successfully in Windows 95. Even DOS programs that require access to all of the system's resources can run successfully in the Windows 95 *MS-DOS mode*. In this mode, basically all but a small portion of Windows exits from memory. When the application is terminated, Windows restarts and returns to the Desktop screen.

MS-DOS mode is establishedfor the application by configuring its properties in the Advanced dialog box under the My Computer/

Properties/*Program tab*. Simply right click on the application's executable file name in the My Computer window and select the MS-DOS Mode setting in the Advanced screen.

It is also possible to adjust the memory allocated to the program through the My Computer/Properties/*Memory tab*. This function is accessed by right clicking its executable file name, moving to the Memory window, and increasing or decreasing the memory available.

The *Memory tab* allows the user to establish memory allocation properties for the application. Values can be selected for Conventional, Extended, and Expanded memory usage, as well as for configuring HMA and UMB operations. These settings are still dependent on the information that may exist in the CONFIG.SYS file. In particular, check the CONFIG.SYS file for noems parameters in the EMM386 statement. If present, replace it with an appropriate **ram** or **x=mmmm-nnnn** parameter.

The *Screen tab* provides several options for how the application will be presented on the screen. It is possible to set the window size that the application will run in. These options include Full Screen, a user-definable window size, and a default window size based on the size determined by the graphic mode the application is using.

This tab also allows the Windows 95 toolbar to be displayed on the bottom of the screen. This feature can be valuable if the application becomes unstable or has trouble running in Windows.

Finally, the Screen tab allows the application to use the Windows 95 Fast ROM emulation and Dynamic Memory allocation features. These functions are selected to speed up video output operations.

Windows 95 Structure

When fully installed, the Windows 95 structure is as depicted in Figure 4.26. The new Registry, Configuration Manager, and Virtual Machine Manager have already been introduced. However, they have been joined by an Installable File System (IFS) Manager to function between the Win95 core and the device drivers that service the system's hardware. On the other side of the Win95 core, applications running on the system are accessed through the new 32-bit, shell and user-interface tools.

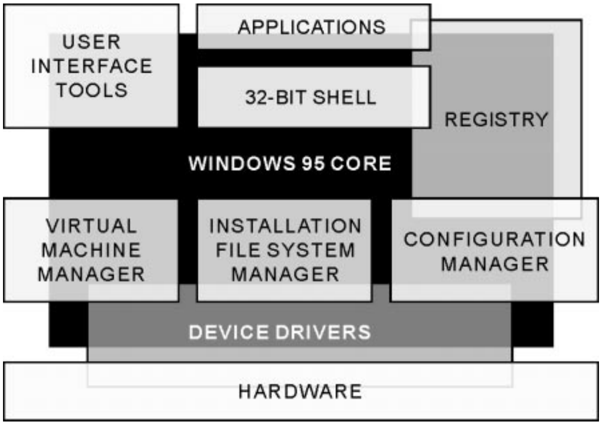


Figure 4.26

The Windows 95 organizational structure.

The Win95 core consist of three components: the Kernel, the GDI, and the USER files. The Kernel is the foundation of the system. It includes basic memory and I/O management, task scheduling, error (exception) handling, and program execution functions. The USER files manage input from hardware devices and output to the user interface components—that is, the icons and screen structures. The GDI components control what appears on the display. It includes two main subsystems: the Graphics subsystem and the Printing subsystem. These components are discussed in more detail later in this chapter.

The Registry

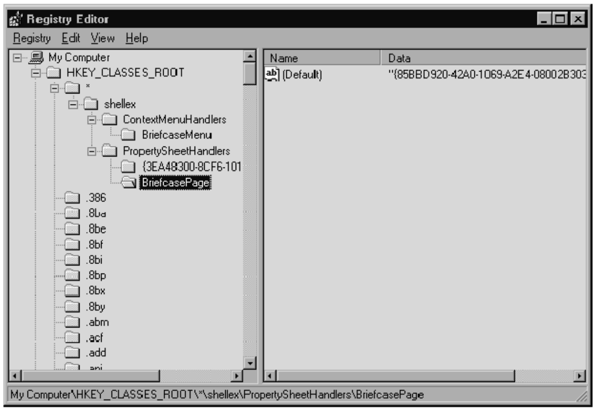
Many of the SYSTEM.INI and WIN.INI management functions of Windows 3.1 have been relocated to an area known as the Registry. The system's configuration information is held in the Win95 Registry. This includes local hardware configuration information, the network environment information, file associations, and user configurations. When applications were removed from the system in earlier Windows versions, the configuration information distributed between the various INI files remained, unless the user, or a special Windows Uninstall program, looked them up and removed them individually. With Windows 95, their headings and the associated configuration information are all removed from the Registry, unlike the old INI method of tracking this information.

The contents of the Registry can be viewed and altered through the Registry Edit utility, shown in Figure 4.27. The Registry files are located in the Windows system directory. Each time Windows 95 boots up successfully, these files are backed with a .DAO extension.

The Registry uses English-language descriptions and a hierarchical organization strategy. The hierarchy is divided into headkeys, keys, subkeys, and values. Keys are descriptive section headers that appear at the left side of the RegEdit window. Values, on the other hand, are definitions of topics organized under the keys. This organization can be thought of in the same terms as the organization of any book; the head keys are similar to chapter titles, the keys and subkeys are equivalent to the major and minor headings of the chapters, and the values are equal to the sentences that convey information.

Figure 4.27

The Win95 Registry Edit window.



Values can contain a wide variety of information types. They can contain interrupt and port address information for a peripheral system, or just information about an installed application program. The information can be encoded into binary, DWORDS, or strings. Values are always located at the right side of the RegEdit window.

If you examine the My Computer heading using the RegEdit option, you will find six categories listed. The head keys all start with an `HKEY_` notation. Under My Computer the categories are `HKEY_CLASSES_ROOT`, `HKEY_CURRENT_USER`, `HKEY_LOCAL_MACHINE`, `HKEY_USERS`, `HKEY_CURRENT_CONFIG`, and `HKEY_DYN_DATA`.

Most of the `HKEY` titles should appear very descriptive of their contents. The `Classes_Root` key divides the system's files into two groups by file extension type and by association. This key also holds data about icons associated with the file. The `Current_User` key holds the data about the user-specific configuration settings of the system including color, keyboard, desktop, and start settings. The values in the `Current_User` key reflect those established by the user who is currently logged on to the system. If a different user logs on, the contents of the `HKEY_USERS` section is moved into the `Current User` key. The `Local_Machine` key contains information about the system's hardware. All the hardware drivers and configuration information is contained in this key. The system cannot use peripheral devices not properly documented in the `Local_Machine` key. Finally, the `Users` key contains the information about the various users that have been defined to log on to the system. The information from the `Current_User` key is copied into this section whenever a user logs off the system, or when the system is shut down.

Windows 95 System Policies

Objectives

Because Windows 95 provides multi-user operations, operational policies are necessary to govern the rights and privileges of different users. Windows 95 `system policies` establish guidelines to restrict user access to the options in the Control Panel and desktop. They also enable an administrator to customize the desktop and configure network settings.

The system policies that govern these functions are established and modified using an editor, similar to the Registry Editor, called the System Policy Editor.

The Policy Editor is another tool that can be used to access the information in the Registry. Unlike the RegEdit utility, the Policy Editor can only access subsets of keys. The Registry Editor can access the entire Registry.

The utility is located on the Windows 95 CD, under the Admin folder so that only the keeper of the CD can adjust the system's policies. The path to access the Policy Editor on the CD is `ADMIN\APPTOOLS\POLYEDIT`. Once located, it can be executed by entering **poledit** in the Run box. This should bring up the Policy Editor screen.

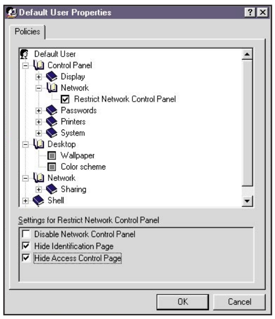
With any multi-user system, it may be necessary to establish various working environments for different users. Some users are entrusted with access to more of the system than other users. As described earlier, this is the purpose of logon procedures. The Win95 policy file tracks policies for different users in a file named CONFIG.POL. The contents of this file are moved into the USER.DAT and SYSTEM.DAT files when a user logs on.

The editor enables system administrators to configure the Windows desktop differently for different users. For some users, it may not be necessary for them to have access to certain system options such as printers or Registry editing tools. Through the Policy Editor, access to these options can be removed from the desktop for a given user.

The window in the Policy Editor screen contains an icon for a default user and a default computer. When the user logs on to the system, Windows searches for a user profile that matches to user logging on. If none is found, the default policies are copied into the new user's profile and used until modified by a system administrator. Figure 4.28 shows the editing screen used to modify the default user's policies.

Figure 4.28

Inside the Policy Editor.



Like the Registry Editor, the branches of the Policy Editor can be expanded or contracted by clicking on the plus and minus signs in the nodes of the tree. Three options can be selected for each setting. These are checked, cleared, or grayed. When a policy is checked, it is being implemented. If it is cleared (open), the policy is not implemented. When the policy is grayed out, the policy has not been changed since the last time the user logged on. As an example of the effects of these settings, use the system's Wallpaper setting from the Control Panel. If the setting is checked, the designated wallpaper will be displayed. If the setting is cleared, no wallpaper will be displayed. If the setting is grayed out, the user can select his own wallpaper pattern through the Control Panel.

Configuration Manager

The Configuration Manager oversees the complete configuration process. Its primary purpose is to ensure that each device in the system can access an interrupt request channel without conflict and that it has a unique I/O port address. The I/O port address is location where the system communicates with an intelligent programmable device. It constantly tracks the number and location of devices in the system and reconfigures them when required.

The Configuration Manager charts a hardware tree for the system similar to the one illustrated in Figure 4.29. The tree represents all the buses and intelligent devices in the system. Information about the buses and devices is collected by the Configuration Manager's bus enumerators. The information can be obtained from the BIOS interrupt services used by the devices, device drivers installed for the devices, and directly from the hardware.

The Configuration Manager assigns the system's resources to the devices. It uses resource arbitrator routines to provide interrupts, DMA channels, I/O addressing, and memory allocations for all the system's devices. The arbitrators resolve any conflicts between the devices, and then informs each device driver about its particular resource allocations.

Virtual Machine Manager

Windows brought multitasking to the personal computer with version 3.0. The system would work its way around all the open applications, allowing them to run for a period of time before resetting and moving to the next application. One of the simplest forms of multitasking is `task-switching`. In a task-switching operation, several applications can be running at the same time. When you have multiple applications open in Windows, the window currently being accessed is called the active window and appears in the foreground (over top of the other windows). The activity of the other open windows is suspended, as denoted by their gray color, and they run in the background.

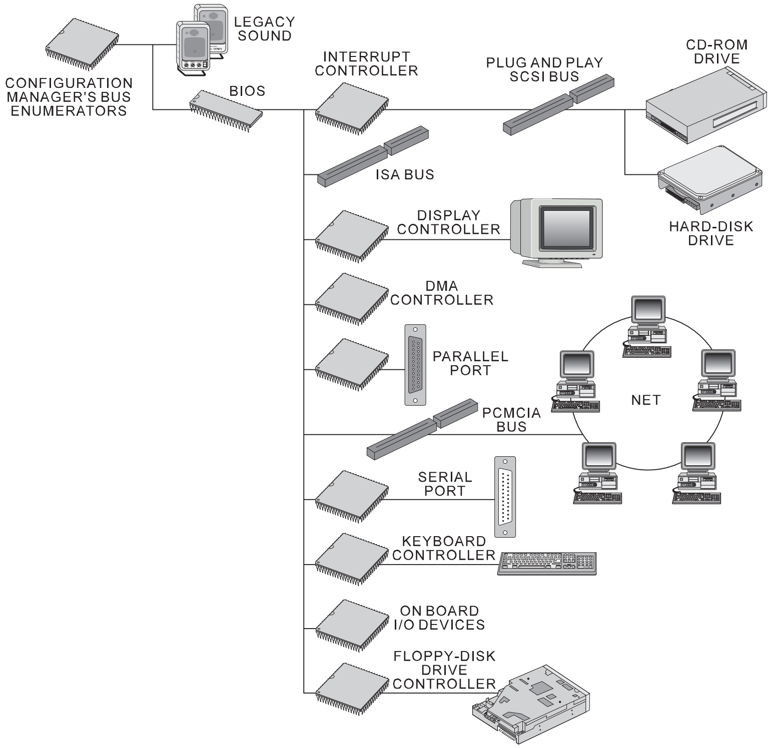


Figure 4.29

The Configuration Manager's tree structure.

Special key combinations enable the user to move between tasks easily. By pressing the Alt and Tab keys simultaneously, you can move quickly through the open applications. The ALT+ESC combination also enables the user to cycle through open application windows.

In a Windows 3.x cooperative multitasking system, some applications gained control of the system and used the resources until they were finished. Some Win 3.x applications took up more than their share of the system's resources.

Microsoft included the preemptive-multitasking operation in the design of Windows 95 so that the operating system allows an application to run for only a predetermined amount of time, based on how critical the application's task is in the overall scheme of the system. More time is allotted to high-priority tasks than to low-priority tasks. However, the operating system remains the controlling force. When the application's time is up, the operating system just cuts it off.

Under cooperative multitasking, the system is tied up with a single application whenever Windows is displaying an hourglass on the screen. With preemptive multitasking, a new task can be opened, or switched to, while the hourglass is being displayed onscreen. Work can be performed under that task window while the other task is being worked on by the system. More importantly, if the system locks up while working on a specific task in Windows 95, you can just end the task rather than restart the machine.

32-Bit Access with Windows 95

Windows 95 streamlines the 32-bit file and disk access operations by removing both the DOS and the BIOS from the access equation, as illustrated in Figure 4.30. This allows Win95 to always run in protected-memory mode; therefore no mode switching needs to occur.

Microsoft refers to this portion of the system as the Protected-Mode FAT File System, or VFAT. As its full name implies, the VFAT provides a protected mode method of interacting with the file system on the disk drive. VFAT operates in 32-bit mode; however, the actual FAT structure of the disk remains as 12- or 16-bit allocations. Because the system does not normally have to exit and reenter protected mode, performance is increased considerably. The logical blocks of the VFAT are described in Figure 4.32.

The VFAT system replaces the SMARTDRV disk caching utility with a protected mode driver named *VCACHE*. Under VCACHE, the size of the cache data pool is based on the amount of free memory in the system instead of a fixed amount. The program automatically allocates blocks of free memory to caching operations as needed. Under Windows 95, the VCACHE driver controls the cache for the system's CD-ROM drive, in addition to the hard disk and file operations. The logical blocks of the VFAT are described in Figure 4.31.

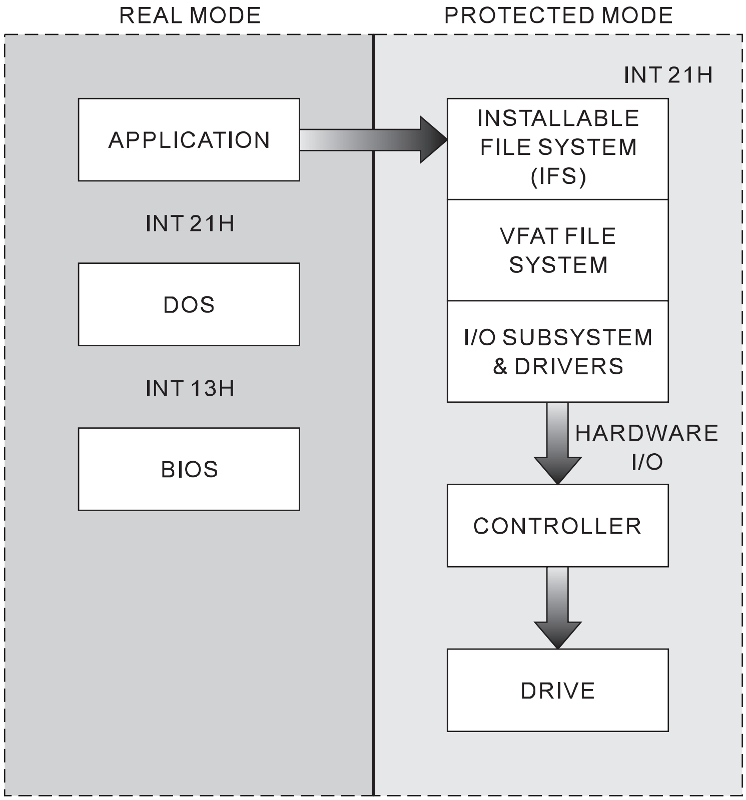


Figure 4.30

32-bit access in Windows 95.

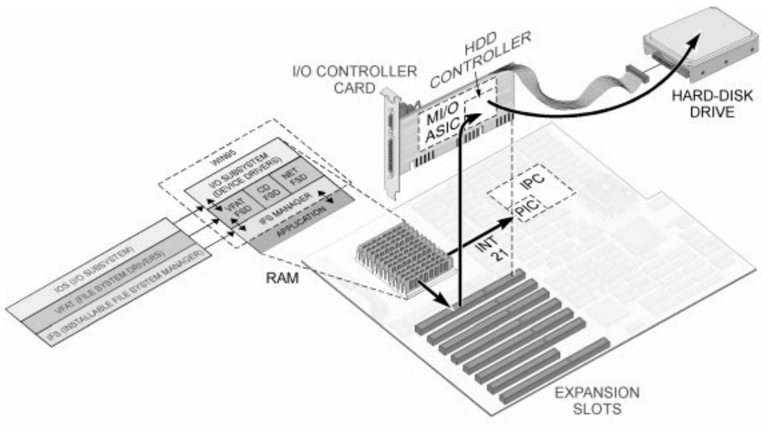


Figure 4.31

The Win95 VFAT interface.

When a file or disk access request is received by Win95, a subsection of the interface known as the installable file system (IFS) processes the request by passing it to the proper file system driver (FSD). The FSDs communicate with the IFS manager and the drivers that work directly with the hardware device controllers. These device-specific drivers work within the I/O subsystem layer (IOS). The IOS layer handles I/O systems that transmit and receive data in multiple-byte transfers. Devices in this category include hard drives, CD-ROM drives, tape drives, and network controllers.

Safe Mode Startup

Objectives

Because there is no DOS level present in the Windows 95 startup routine, special precautions and procedures must be used to protect the system in case of startup problems.

Two very good tools to use in these situations are an Emergency Startup Disk and the Startup menu.

During the Windows 95 setup, the software provides an option for creating an Emergency Startup Disk. This option should be used for every Windows 95 installation. Setup copies the operating system files to the disk along with utilities for troubleshooting startup problems. The disk can then be used to boot up the system in Safe mode and display a DOS command-line prompt.

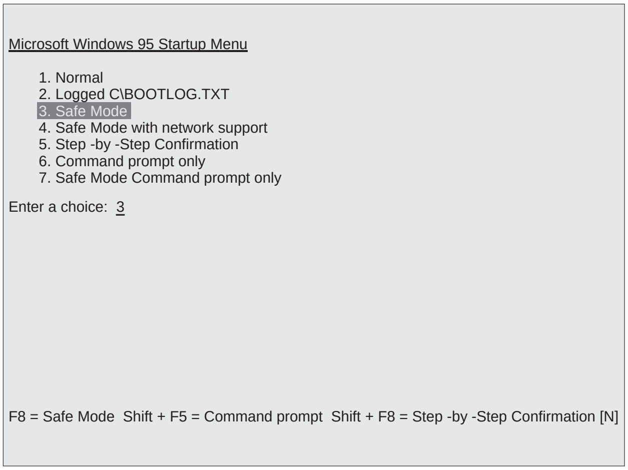
A startup disk can also be created through the Control Panel's Add/Remove Programs icon. This option is normally used to create a new startup disk after new hardware has been installed or configuration information has been changed.

The Windows 95 Startup menu, shown in Figure 4.32, can be obtained on a nonstarting system by holding the F8 function key when the Starting Windows 95 display is onscreen. The menu offers several startup options, including: Normal, Logged, Safe, and Step-by-Step Confirmation modes.

In Normal mode, the system just tries to restart as it normally would, loading all its normal startup and Registry files. The Logged mode option also attempts to start the system in Normal mode, but keeps an error log file that contains the steps performed and outcome. This text file (BOOTLOG.TXT) can be read with any text editor, or printed out on a working system.

Figure 4.32

The Startup menu.



If Win95 determines that a problem has occurred that prevented the system from starting, it attempts to restart the system in Safe mode. This mode bypasses several startup files to provide access to the system's configuration files. In particular, the CONFIG.SYS and AUTOEXEC.BAT files are bypassed, along with the Win95 Registry and the SYSTEM.INI's `[Boot]` and `[386enh]` sections. In this mode, the keyboard, mouse, and Standard-mode VGA drivers are active. Unless modified, the Safe Mode screen appears as depicted in Figure 4.33. Active functions appear on the screen along with the Safe Mode notice in each corner.

Figure 4.33

The Safe Mode startup screen.

Safe mode can also be accessed by typing `Win /d:m` at the DOS prompt, or by pressing the F5 function key during startup.

The Step-by-Step Confirmation mode displays each startup command line-by-line and waits for a confirmation from the keyboard before moving ahead. This allows an offending startup command to be isolated and avoided so that it can be replaced or removed. This option is obtained by pressing the F8 function key at the Startup menu.

Other startup options may also be available from the menu depending on the configuration of the system. Some options start the system and bring it to a DOS command-line prompt. Depending on which option is selected, the system may boot up to the command line, using the startup files and the Registry, or start in Safe mode with a command-line prompt only.

Windows can be started from the command line using a number of different switches. These switches can be used to check for startup conflicts. The `Win /d:m` example given earlier starts Windows in Safe mode. The `/d:x` switch is used to check for an upper-memory conflict. The `/d:f` switch can be used to check for 32-bit disk access conflicts. Finally, the `/d:v` switch is used to check for hard disk I/O conflicts.

Windows 95 maintains a number of log files that track system performance and can be used to assess system failures. These log files are SETUPLOG.TXT, DETLOG.TXT, and BOOTUPLOG.TXT and are stored in the drive's root directory. All three are text files that can be viewed with a text editor package.

Their filenames are indicative of the types of information they log. As described earlier, the BOOTUPLOG.TXT file tracks the events of the Startup procedure. Likewise, SETUPLOG.TXT tracks the events of the Setup process. The DETLOG.TXT file monitors the presence of detected hardware devices and identifies the parameters for them.

Windows 95 - OSR2

Windows 95 OSR2, also known as Window 95, is an upgrade of the original Windows 95 package that includes patches and fixes for version 1, along with Microsoft Internet Explorer 3.0 and Personal Web Server. It also includes an enhanced file allocation table system referred to as FAT32.

Previous versions of DOS and Windows supported what is now termed FAT16 (or FAT12). As described in Chapter 3, the size of the operating system's FAT determines the size of the clusters for a given size disk partition. Of course, smaller cluster sizes are better because even a single byte stored in a cluster will remove the entire cluster from the available storage space on the drive. This can add up to a lot of wasted storage space on larger drives. Table 4.2 describes the relationships between clusters and maximum partitions for various FAT entry sizes.

Table 4.2

FAT Relationships

FAT Type	Partition Size	Cluster Size (In bytes)
FAT12	16 MB	4096
FAT16	32 MB	2048
FAT16	128 MB	2048
FAT16	256 MB	4096
FAT16	512 MB	8192
FAT16	1 GB	16384
FAT16	2 GB	32768
FAT32	<260 MB	512
FAT32	8 GB	4096
FAT32	16 GB	8192
FAT32	32 GB	16384
FAT32	>32 GB	32768

To use the FAT32 system, the hard drive must be formatted using the FDISK/FORMAT functions in OSR2. This makes FAT32 incompatible with older versions of Windows (even Windows 95 and Windows NT) and with disk utilities and troubleshooting packages designed for FAT12/16 systems.

To use the FAT32 FDISK function in OSR2, it is necessary to enable the Large Disk Support option. After completing the FDISK function and exiting, it is necessary to manually reboot the system. After this, it is usually a simple matter of performing a FORMAT operation using the OSR2 CD, or Start Disk, to install the FAT32 drive. Failure to reboot between the FDISK and FORMAT operations will produce an error.

To verify that the hard drive is formatted with FAT32, select the My Computer option from the desktop and right-click on the C: drive icon. This will produce the `[C:]` properties window displayed in Figure 4.34. The Type entry should read `Local Disk [FAT32]`. The hard disk usage pie chart will not work correctly with drives larger than 2 GB. It will show the drive as empty until at lease 2 GB of space is used.

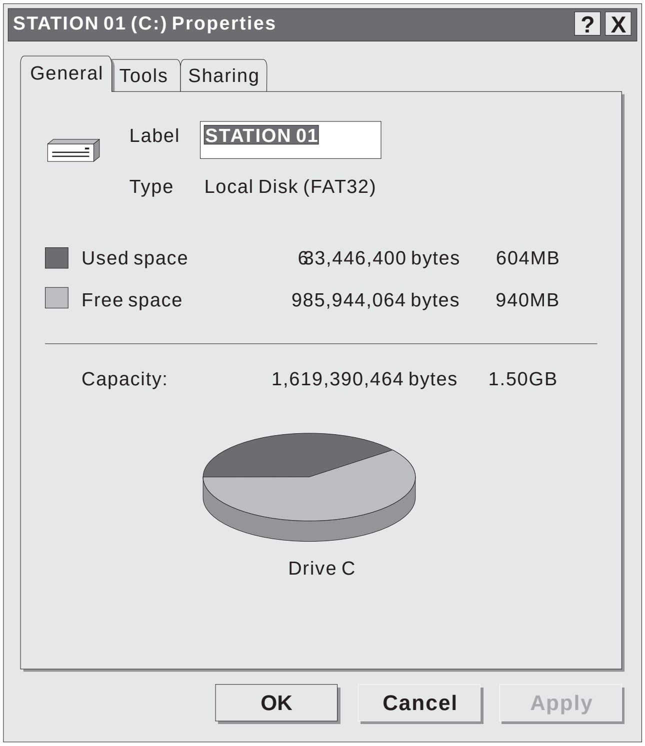


Figure 4.34

Showing FAT32 in the HDD Properties window.

OSR2 does not require that FAT32 be used. It will operate just as well, if not better, using the FAT16 format. Depending on the application of the system, it may run slower with FAT32. Remember that FAT 32 is designed to optimize storage space, not performance. The simple fact that FAT32 offers more the potential for more clusters makes it slower than a drive with less clusters. With this in mind, the decision to use of FAT32 or FAT16, or to use different cluster sizes in FAT32, usually depends on the balance the user establishes between performance and storage. The default cluster size set by Microsoft for FAT32 is 4 KB.

In addition to the FAT32 system, OSR2 offers improved power management (APM) functions, bus mastering support, MMX multimedia support, and enhanced PCMCIA functions over version (which is referred to as OSR1, or Service Pack 1).

Also new in OSR2 is HDD/CD-ROM DMA access support. This feature is located in the Control Panel/System/Device Manager/Disk Drives window. At this point, choose the desired drive, select Properties, and click the Settings tab. Check the DMA box and reboot the system. The same procedure should be performed for the CD-ROM drive as well. This box will only appears for IDE drives, and only when using properly installed and configured OSR2 bus mastering drivers for the drive.

Interestingly, OSR2 is not sold on the retail market. It is only available legally through OEMs that are selling system board and HDD hardware or with new systems. Version can be identified by 0796 Part No. 000-45234 nomenclature on the CD, or by a B in the part number under the System Properties tab. It may also be denoted by a 4.00.950.1111 version number when a VER operation is performed.

Summary

This chapter has presented a foundation for the Windows 3.x operating environment and the Windows 95 operating system. These system software programs are the foundation for almost everything that goes on in personal computer system. As the rest of the chapters in this book focus in on particular hardware systems, it is important to remember that Windows is somewhere (usually in the background) and will need to be contended with as part of a troubleshooting process for the component. Advance Windows concepts and component-specific Windows information is presented with each of the components covered.

Lab Exercise

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the lab manual and perform Procedures 3 — Windows, 4 — Advanced Windows, 5 — Windows 95, and 6 — Advanced Windows 95.

Review Questions

1. What are the maximum and minimum color value settings for Windows screen components?
2. How many parallel ports does Windows recognize?
3. What type of multitasking is performed in Windows 3.x Standard mode operation?
4. What type of multitasking is performed in Windows 3.x 386 Enhanced mode operation?
5. What is a PIF file, and what does it do?
6. Which Windows utility is used to edit ASCII files, such as COMMAND.COM and the INI files?
7. Describe the basic responsibilities of the INI files in Windows 3.x.
8. The minimum amount of extended memory required to start Windows in Enhanced mode is _________.
9. Why are logon procedures used and where?
10. What function does the BOOTUPLOG.TXT file serve?
11. Name two important tools for solving startup problems in Windows 95.
12. What practical function does 32-bit access serve in the Windows 95 system?
13. What is the significance of placing items in the Windows Startup window?

14. What type of multitasking is performed in Windows 95 386 Enhanced mode operation? Why is it different than the 3.x multitasking operation?

15. How are the branches of a Win95 Registry or policy tree expanded? Contracted?

16. Using the System Policy Editor, what effect does a grayed out setting have on an option?

17. The file that loads the Windows 3.x environment is _________.

18. List two methods of performing a Safe mode startup in Windows 95.

19. What INI functions does the Windows 95 Registry file take over from Windows 3.x, and why?

20. Why would a Step-by-Step Confirmation mode startup be performed?

21. How are the `Setup`, `Setup/A`, and `Setup/I` commands different?

22. How are the `Win`, `Win/s`, and `Win/3` commands different?

23. What is the major input device for a Windows-based system?

24. What are the requirements to start Windows in 386 Enhanced mode?

25. Where is the Windows 95 Policy Editor located?

Review Answers

1. The range of possible values runs from 0 to 255; 0 is minimum intensity, and 255 is maximum intensity. For more information, see the section titled "The `[Colors]` Section."

2. Windows recognizes up to three parallel ports and assigns them LPTx designations. For more information, see the section titled "The `[Ports]` Section."

3. Although it was possible to task-switch in Standard mode, multitasking was not possible. For more information, see the section titled “Running Windows.”

4. In Windows 386 Enhanced mode, the operating system runs in Cooperative Multitasking mode. The various tasks cooperate with each other, more or less, in sharing control of the system. For more information, see the section titled “DOS and Windows.”

5. PIFs are program information files created for any DOS-based files Setup finds in the system. For more information, see the section titled “Installing Windows 3.x.”

6. The Notepad utility. For more information, see the section titled “Initialization (INI) Files.”

7. The Windows INI files hold the default and current startup settings for the various Windows components. For more information, see the section titled “Initialization (INI) Files.”

8. Windows 3.1 requires 2 MB, and Windows 3.11 needs 4 MB. For more information, see the section titled “Installing Windows 3.x.”

9. The logon process can be used to protect the system from unauthorized users. It also allows the system to create a custom environment for each user who may work at any given station. Information specific to a particular user can be stored in the system and recalled when that person logs on. For more information, see the section titled “Security.”

10. The BOOTUPLOG.TXT file tracks the events of the startup procedure. For more information, see the section titled “Safe Mode Startup.”

11. Two very good tools to use in these situations are an Emergency Startup Disk and the Startup menu. For more information, see the section events of the startup “Safe Mode Startup.”

12. 32-bit access allows Win95 to always run in protected-memory mode; therefore no mode switching needs to occur. For more information, see the section titled "32-Bit Access with Windows 95."

13. A programs can be set up to automatically start along with Windows by placing its icon in the Startup window. For more information, see the section titled "Safe Mode Startup."

14. Windows 95 employs preemptive multitasking. Under cooperative multitasking, the system is tied up with a single application whenever Windows is displaying an hourglass onscreen. With preemptive multitasking, a new task can be opened, or switched to, while the hourglass is being displayed onscreen. For more information, see the section titled "Virtual Machine Manager."

15. The branches of the Registry or Policy Editors can be expanded or contracted by clicking on the plus and minus signs in the nodes of the tree. For more information, see the section titled "Windows 95 System Policies."

16. When the policy is grayed out, the policy has not been changed since the last time the user logged on. For more information, see the section titled "Windows 95 System Policies."

17. WIN.COM. For more information, see the section titled "Running Windows."

18. Safe mode can also be accessed by typing **Win /d:m** at the DOS prompt, or by pressing the F5 function key during start-up. For more information, see the section titled "Safe Mode Startup."

19. The Win95 Registry takes over many of the SYSTEM.INI and WIN.INI management functions. This file consolidates configuration information about different applications into a single file so that they can be uninstalled more easily. For more information, see the section titled "The Registry."

20. The Step-by-Step Confirmation mode displays each startup command line-by-line and waits for a confirmation from the keyboard before moving ahead. This allows an offending startup command to be isolated and avoided, so that it can be replaced or removed. For more information, see the section "Safe Mode Startup."

21. A standard setup just involves placing the Setup Disk 1 in the A: drive, and typing **SETUP** at the A:\>> prompt. To install a shared copy of Windows on a network drive, type **SETUP/A**. Likewise, if you type **SETUP/I** Windows will start without regards to any automatic hardware detection. For more information, see the section titled "Installing Windows 3.x."

22. The Windows program can be loaded into the system by typing **WIN** at the C:\>> prompt. Typing **WIN/S** forces the system to start Windows in Standard mode. Typing **WIN/3** forces the system to run in Enhanced 386 mode, if possible. For more information, see the section to install a shared copy of Windows on a network drive "Running Windows."

23. The mouse. For more information, see the section titled "Running Windows."

24. To start Windows in 386 Enhanced mode, a minimum of an 80386SX microprocessor, 256 KB of free conventional memory, and an extended memory of 1024 KB are required. A memory manager must also be loaded ahead of Windows. For more information, see the section titled "Installing Windows 3.x."

25. The utility is located on the Windows 95 CD, under the Admin folder so that only the keeper of the CD can adjust the system's policies. The path to access the Policy Editor on the CD is `ADMIN\APPTOOLS\POLYEDIT`. For more information, see the section titled "Windows 95 System Policies."

Chapter 5

Troubleshooting the System

Upon completion of this chapter and its related Lab Procedures, you should be able to:

Objectives

- Describe the characteristics of a good workspace.
- Outline steps for using a digital multimeter to perform voltage, resistance, and current checks on a system. Identify common DMM tests associated with personal computers.
- List preliminary steps for diagnosing computer problems.
- Describe the three general categories of problems into which symptoms can be grouped and differentiate among them.
- Differentiate between software- and hardware-based troubleshooting techniques.
- Describe the function of a POST card.
- Describe quick checks that can be used to determine the nature of system hardware problems.
- List tools associated with FRU troubleshooting techniques.
- Describe FRU-level troubleshooting.
- Discuss methods of dealing with symptoms that are not defined well enough to point to a particular component.

continues

- Perform visual inspections of a system.
- Use disk-based diagnostic tools to isolate system problems.
- Use DOS batch files to help test selected areas of the system.
- Perform checks to isolate problems that produce a Dead system.
- Create a `clean` boot disk for troubleshooting Windows 3.x problems.
- Use the DRWATSON file to monitor Windows 3.x problems.
- Create an Emergency Start disk for troubleshooting Windows 95 startup problems.
- Use log files to determine the location of Windows 95 operating system problems.

Introduction

Effective troubleshooting of electronic equipment is a matter of combining good knowledge of the equipment and its operation with good testing techniques and deductive reasoning skills. In general, the process of troubleshooting microprocessor-based equipment begins at the outside of the system and moves inward. The first step is always to try the system to see what symptoms are produced. Second, you must isolate the problem to either software- or hardware-related problems. After this, the problem should be isolated to a section of the hardware or software. Finally, the problem must be isolated to the offending component.

The information in this chapter instructs you on the theory behind successful troubleshooting tools and methods you need to use to effectively troubleshoot microprocessor-based equipment.

Tools and Workspace

Objectives

The first order of business when working on any type of electronic equipment is to prepare a proper work area.

You need a clear, flat workspace on which to rest the device. Make sure your workspace is large enough to accommodate the work piece. Check to make sure that an adequate number of power receptacles are available to handle all the equipment you may need. Try not to locate your workspace in a high-traffic area. Try to put it somewhere where no one will notice if it's there for a few days.

Good lighting is a prerequisite for the work area because the technician must be able to see small details such as part numbers, cracked circuit foils, or solder splashes. An adjustable lamp with a shade is preferable. Fluorescent lighting is particularly desirable. In addition, a magnifying glass can prove to be a helpful item when trying to read small part numbers or when looking for cracks in printed circuit board traces.

Organizational Aids

Because some troubleshooting problems may require more than one session, it's a good idea to have some organizational aids in hand before you begin to disassemble any piece of equipment. The following are some of the organizational aids you need:

- A parts organizer to keep track of small parts such as screws and connectors you may remove from the device. This organizer need not be extravagant. A handful of paper or Styrofoam cups does nicely, as does a handful of clear plastic sandwich bags.
- A roll of athletic or masking tape. The tape can be used to make tags and labels to help identify parts, where they go, and how they are connected in the circuit. The worst thing you can possibly do is attempt to remember everything in your head. Take the time to write notes and stick them on your parts organizers, circuit boards, and cables you remove from the system.
- A small notepad or notebook to keep track of your assembly/troubleshooting steps. By the time you begin to disassemble a unit, there should already be quite a few entries logged explaining what preliminary steps led to the decision to remove the outer cover of the device.

Diagnostic and Repair Tools

Obviously, anyone who's going to work on any type of equipment must have the proper tools for the task. This section discusses the tools and equipment associated with the testing and repair of digital systems.

Using Hand Tools

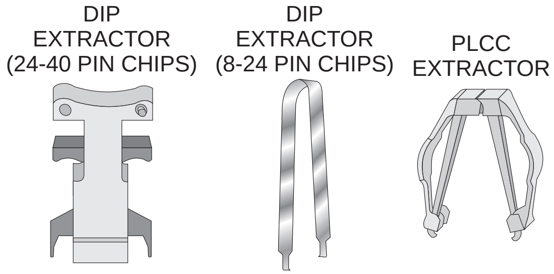
First, you need some common hand tools. The well-prepared technician's tool kit should contain a wide range of both flat-blade and Phillips-type screwdriver sizes. At a minimum, it should have a small jeweler's and a medium-sized flat-blade screwdriver, along with a medium-sized Phillips screwdriver. In addition, you may want to include a small set of miniature nut drivers, a set of Torx drivers, and a special nonconductive screwdriver-like device called an `alignment tool`.

You also need a couple of pairs of needle-nose pliers. These pliers are available in a number of sizes. You need at least one pair with a sturdy, blunt nose and one with a longer, more delicate nose. You may want to get a pair which also has a cutting edge built into its jaws. This same function may be performed by a different type of pliers called `diagonals`, or cross-cuts. Many technicians carry a pair of surgical forceps in addition to their other pliers.

Another common set of tools associated with computer repair are IC pullers, or IC extractors. These tools come in various styles, as illustrated in Figure 5.1, and are used to remove ICs from sockets. Socket-mounted ICs are not as common on modern PC boards as they were in the past. Potential failures associated with the mechanical connections between sockets and chips, coupled with the industry's reliance on surface-mount soldering techniques, have led to far fewer socket-mounted chips. However, there are still occasions, such as when upgrading a ROM BIOS chip, where the IC puller comes in handy.

Figure 5.1

IC pullers.



A desk-mount vise, or a specialized tool to hold the work piece (printed circuit boards and so forth) steady during testing and repair, is a valuable asset. Hand tools commonly associated with microcomputer repair are depicted in Figure 5.2.

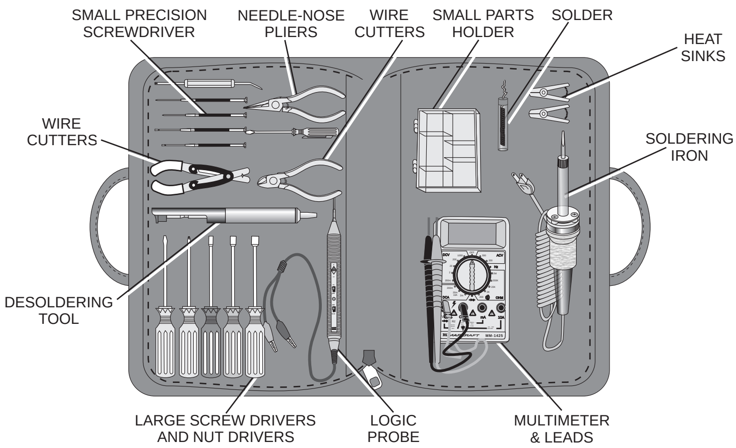


Figure 5.2

Hand tools.

Using a Multimeter

Objectives

A number of test instruments can be most helpful in isolating problems. One of the most basic pieces of electronic troubleshooting equipment is the `multimeter`. These test instruments are available in both analog and digital read-out form and can be used to directly measure values of `Voltage` (V), `Current`, in milliamperes (mA) or amperes, and `Resistance`, in Ohms (Ω). Therefore, these devices are referred to as `VOMs` (Volt-Ohm-Milliammeters) for analog types, or `DMMs` (Digital MultiMeters) for digital types. Figure 5.3 depicts a Digital Multimeter.

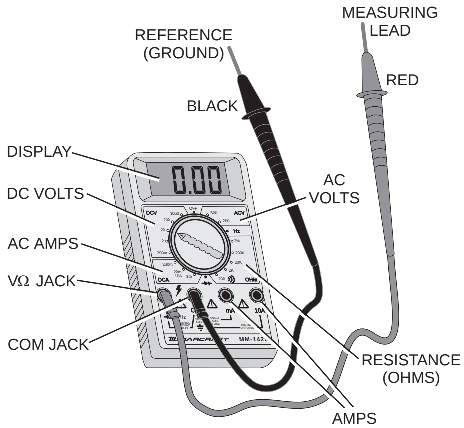


Figure 5.3

Digital multimeter.

With a little finesse, this device can be used to check diodes, transistors, capacitors, motor windings, relays, and coils. The DMM illustrated in Figure 5.3 contains facilities built into the meter to test transistors and diodes. This is in addition to its standard functions of current, voltage, and resistance measurement.

The first step in using the multimeter to perform tests is to select the proper function. For the most part, you never need to use the current functions of the multimeter when you are working with computer systems. However, the voltage and resistance functions can be very valuable tools.

In computer and peripheral troubleshooting, fully 99% of the tests made are DC voltage readings. These measurements most often involve checking the DC side of the power supply unit. These readings can be made between ground and one of the expansion slot pins (see Chapter 6, "System Boards," for expansion slot specifications) or at the P8/P9 power supply connectors. It is also common to check the voltage level across a system-board capacitor to verify that the system is receiving power. The voltage across most of the capacitors on the system board is 5VDC. The DC voltages that can normally be expected in a PC-compatible system are +12V, +5V, -5V, and -12V. The actual values for these readings may vary by five percent in either direction.

The DC Voltage Function is used to take measurements in live DC circuits. It should be connected in parallel with the device being checked. This could mean connecting the reference lead (black lead) to a ground point and the measuring lead (red lead) to a test point to take a measurement, as illustrated in Figure 5.4. It is normal practice to first set the meter to its highest voltage range to make certain that the voltage level being measured does not damage the meter. As an approximate value is detected, the range setting can be decreased to achieve a more accurate reading. Most meters have over-voltage protection. However, it is still a good safety practice to decrease the range of the meter after an initial value has been achieved.

The second most popular test is the resistance, or continuity test. Unlike the voltage check, resistance checks are always made with power removed from the system. Failure to turn off the power when making resistance checks can cause serious damage to the meter and pose a potential risk to the user. Resistance checks also require that the component being tested be electrically isolated from the system. For most circuit components, this means desoldering at least one end from the board.

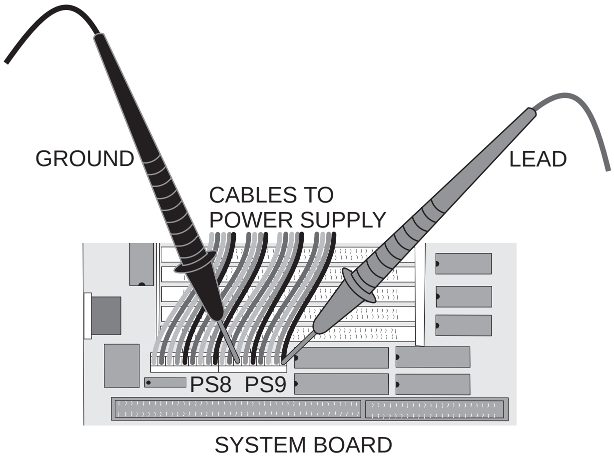


Figure 5.4

DC voltage check.

However, the resistance check is very useful in isolating some types of problems in the system. One main use of the resistance function is to test fuses. At least one end of the fuse must be disconnected from the system. The meter should be set on the 1k-ohm resistance setting. If the fuse is good, the meter should read near zero ohms. If it is bad, the meter reads infinite. The resistance function is also useful in checking for cables and connectors. If you remove the cable from the system and connect a meter lead to each end, you can check the cable's continuity conductor-by-conductor to verify its integrity. The resistance function is also used to test the system's speaker. To check the speaker, simply disconnect the speaker from the system and connect a meter lead to each end. If the speaker is good, the meter should read near zero ohms. If the speaker is defective, the resistance reading should be infinite.

Only a couple of situations involve using the `AC voltage function` for checking microcomputer systems. The primary use of this function is to check the commercial power being applied to the power supply unit. As with any measurement, it is important to select the correct measurement range. However, the lethal voltage levels associated with the supply power call for additional caution when making such measurements. The second application for the AC voltage function is to measure ripple voltage from the DC output side of the power supply unit. This particular operation is very rarely performed in field service situations.

Initial Troubleshooting Steps

Objective

As a general rule, the majority of all equipment problems can be reduced to the simplest things you can think of. The problem is, most people don't think of them. Successful troubleshooting is the result of careful observation, deductive reasoning, and an organized approach to solving problems. These techniques are common to repairing any type of defective equipment. Although the demonstration of these techniques is applied to repairing computer systems, it is quite possible to adapt them to other systems as well.

The most important thing to do when you approach a malfunctioning device is to be observant. Begin your efforts by talking to the person who reported the problem. Many clues to a problem can be determined from this person. Also be aware that many problems are actually related to the operator. One of the first things to do if you are not personally familiar with the system is to eliminate the user as a possible source of the problem.

List the procedures that led up to the malfunction. This communication can help you narrow a problem down to a particular section of the computer. It does no good to check the video display when the user is having trouble using the disk drive. Next, observe the symptoms of a malfunction to verify the problem for yourself. After a problem has been identified, try to associate the malfunction with a section of the system responsible for that operation.

Performing the Visual Inspection

Objective

If no one has prior knowledge of the type of malfunction, you should proceed by performing a careful visual inspection of the system. Check the outside of the system first. Look for loose or disconnected cables. Consult all the external front-panel lights. If no lights are displayed, check the power outlet, the plugs and power cords, as well as any power switches that may affect the operation of the system. You may also want to check the commercial power-distribution system's fuses or circuit breakers to see that they are functional.

If part of the system is active, try to *localize the problem* by systematically removing peripheral devices from the system. Try swapping out suspected devices with known good parts from another computer of the same type. Try to revive the system, or its defective portion, by restarting it several times. As a matter of fact, you should try to restart the system after each correctional step is performed.

Check all externally accessible switch settings. For example, check all system jumper settings to see that they are set correctly for the actual configuration of the system. In Pentium-based systems, check the BIOS Advanced CMOS Configuration screen for enabling settings that may not be correct. Also, make certain that any peripheral devices in the system, such as printers or modems, are set up correctly. Consult any additional users' or operations manuals liberally. Indeed, many of the computers and peripheral systems on the market, such as printers, have some level of self-diagnostics built into them. Generally, these diagnostics programs produce coded error messages. The key to recognizing and using these error messages is usually found in the device's user's manual. In addition, the user's manual may contain probable causes and suggested remedy information, or specialized tests to isolate specific problems.

Note

> Take the time to *document the problem*, including all the tests you perform and their outcomes. Your memory is never as good as you think it is, especially in stressful situations such as with a down computer. This recorded information can prevent you from making repetitive steps, which waste time and may cause confusion. This information is also very helpful when you move on to more detailed tests or measurements.

Watching the Bootup Procedure

Carefully watching the steps of a bootup procedure can reveal a lot about the nature of problems in a system. Faulty areas can be included or excluded from possible causes of errors during the bootup process. The observable actions of a working system's cold-boot procedure are listed as follows, in their order of occurrence:

1. When power is applied, the power supply fan should work.
2. The keyboard lights should flash as the rest of the system components are being reset.
3. A BIOS message should be visible on the monitor.
4. A memory test should be visible on the monitor.
5. The floppy-disk drive access light should come on briefly.
6. The hard-disk drive access light should come on briefly.
7. An audible beep should be heard.
8. The floppy-disk drive access light should come on briefly before switching to the hard drive.
9. For DOS-based machines, a DOS prompt should be visible on the monitor. For Windows 95 machines, the message Starting Windows appears on the screen.

If a section of the computer is defective, some or none of these steps are observed. If you know the sections of the computer involved in each of the steps, you may safely suspect a particular section of causing the problem if the system does not advance past that step. As an example, it would not be logical to replace the floppy-disk drive (5) when a memory test (4) has not been observed on the monitor.

When a failure occurs, you can eliminate components as a possible cause by observing the number of steps that the system completes in the preceding list. Those subsystems associated with steps successfully completed can be eliminated. Efforts should be focused on only those sections responsible for the symptom. When that symptom is cleared, the computer should progress to another step. However, there still may be another unrelated symptom to appear farther down the list. This symptom should be dealt with in the same manner. Always focus on diagnosing the present symptom and eventually all the symptoms will disappear.

Determining Hardware/Software/ Configuration Problems

Objective

It should be obvious that a functional computer system is composed of two major parts: the system's hardware and the software that controls it. These two elements are so closely related that it is often difficult to determine which part might be the cause of a given problem. Therefore, one of the earliest steps in troubleshooting a computer problem (or any other programmable system problem) is to determine whether the problem is due to a hardware failure or to faulty programming.

The easiest way to determine whether a problem is hardware- or software-related is to test the hardware with software packages that are known to be good and that have successfully run on the system before. If the system boots up properly, and runs known, good programs correctly, then the problem is very likely to be software-related. If the system does not boot, or refuses to run programs that previously ran on it, then the problem is likely to be hardware-related.

Note

> The majority of all problems that occur in computer systems are in the area of software and configuration settings.

There's a special category of problems that tend to occur whenever a new hardware option is added to the system, or when the system is used for the very first time. These problems are called `setup problems`, and are due to mismatches between the system's programmed configuration and the actual equipment installed in the system. This mismatch can also be between the system's configuration settings and hardware jumper and switch settings.

It is normally necessary to run the system's CMOS Setup utility in three situations:

> The first situation occurs when the system is installed for the first time.
>
> The second situation occurs if it becomes necessary to replace the CMOS backup battery on the system board.
>
> Finally, any time a new option is added to the system, it may be necessary to run Setup.

These options normally include the disk drive, the video display, and the installed memory.

`Configuration problems` occur with some software packages when they are first installed. Certain parameters must be entered into the program by the user to match its capabilities to the actual configuration of the system. These configuration settings are established through the startup software in the ROM BIOS. If these configuration parameters are set incorrectly, the software is unable to direct the system's hardware properly and an error occurs.

When you are installing new hardware or software options, be aware of the possibility of this type of error occurring. If configuration, or setup, errors are encountered, refer to the installation instructions found in the new component's user's manual. Table 5.1 lists typical configuration error codes and messages produced when various types of configuration mismatches are incurred.

Table 5.1

Common configuration error codes.

Configuration Error Message	Meaning
CMOS System Option Not Set	Indicating failures of CMOS battery or CMOS checksum test.
CMOS Display Mismatch	Failure of display type verification.

continues

Table 5.1 Continued

Configuration Error Message	Meaning
CMOS Memory Size Mismatch	System configuration and setup failure.
Strike F1 to Continue	Invalid configuration information.

Software Diagnostics

Most PCs have reasonably good built-in self-tests that are run each time the computer is powered up. These tests can prove very beneficial in detecting hardware-oriented problems within the system. Whenever a self-test failure occurs, the system may indicate the error through an audio response (beep codes), a blank screen, or a visual error message on the video display. Some PCs issue a numerically coded error message on the display when an error occurs. However, other PCs display a written description of the error.

Basically, software diagnostic routines check the system out by running predetermined sets of tests on different areas of the system's hardware. The diagnostic package evaluates the response from each test and attempts to produce a status report for all the system's major components. Like the computer's self-test, these packages produce visual and beep-coded error messages.

The most common software troubleshooting packages test the system's memory, microprocessor, keyboard, display monitor, and the disk drive's speed. If at least the system's CPU, disk drive, and clock circuits are working, you may be able to use one of these special software troubleshooting packages to help localize system failures. They can be especially helpful when trying to track down non heat-related intermittent problems. However, these test programs are effective down to the board level only, not down to a particular chip on a particular board.

If the diagnostic program indicates that multiple items should be replaced, replace the units one at a time until the unit starts up. Then, replace any units removed prior to the one that caused the system to start. This ensures that there were not multiple bad parts. If all the parts have been replaced, and the unit still does not function properly, the diagnostic software is suspect.

Using ROM-Based Diagnostics

In cases where the system is unable to load information from a floppy disk, ROM-based diagnostic programs are available that can be used to check out the partially dead system. In other systems, the diagnostic chips are usually substituted for the system board's ROM BIOS chip.

Several companies offer disk-based and ROM-based diagnostics packages for troubleshooting computer problems. Some are better than others. As a group, ROM-based diagnostics packages are better than disk-based packages simply because they require much less of the system to be operable for use. Disk-based programs require that almost all the system be functional before they can be used.

Some versions of the AMI BIOS come with an extensive set of diagnostic routines built into the ROM package. Therefore, you can use the diagnostic routines without removing the system unit's outer cover. These routines can be entered through the BIOS' CMOS Setup menu. To invoke the diagnostic program, turn on the computer and wait for the message "Press DEL if you want to run SETUP or DIAGS" and press the DEL key. After entering the CMOS Setup screen, simply move the screen cursor to the `Run Diagnostics` position and press the ENTER key. The BIOS routine responds by placing its diagnostics selection menu, illustrated in Figure 5.5, on the monitor screen. You select the section to test, along with the specific test to run, by using the ARROW keys.

The diagnostic menu screen is divided into three major portions:

- AMI ROM Diagnostics header
- Field for listing available procedures and tests
- List of present devices known to the system

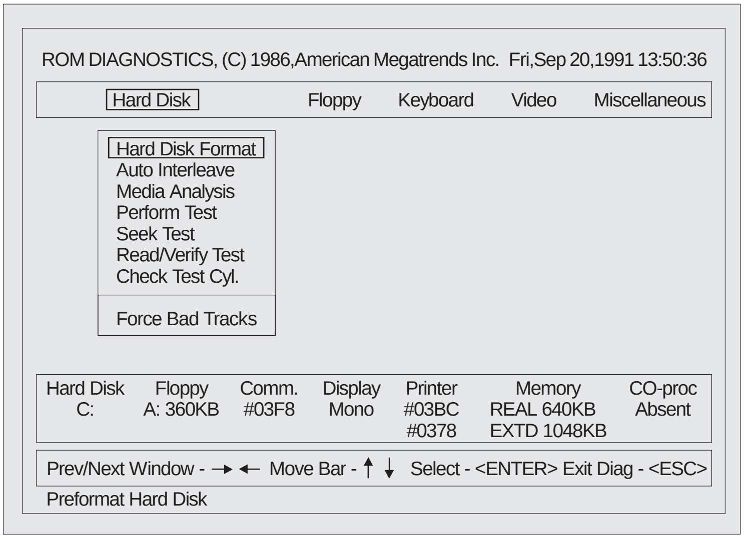


Figure 5.5

The CMOS diagnostics menu.

At the top of the screen is the AMI ROM Diagnostics header. This header contains the BIOS version number, the computer's current time and date information, and five headings that can be selected for diagnostic operations. These topics include Hard-Disk Drive procedures and test functions, Floppy-Disk Drive procedures and test functions, Keyboard tests, Video display tests, and Miscellaneous port tests.

When the highlighted cursor is moved to one of the header topics, a listing of the available procedures and tests appears in the section of the screen underneath the topic. At the bottom of the screen, the program displays a list of the devices that it knows are present in the system. Directions for navigating through the program are also presented at the bottom of the page.

When you select a test listed below one of the diagnostic topics, firmware test routines stored in the ROM BIOS are executed. These diagnostic programs can be used to determine the functionality of the computer's sub-systems. The routines are menu-driven and enable you to select a specific test to execute. They also provide information about devices installed in the system.

The Printer Adapter test checks the signals of the parallel port. During the test, a printer should be connected and the power should be ON. This test reports any abnormalities in the printer port status signals, such as whether the printer is loaded with paper. The `parallel loopback plug` is normally used to simulate the presence of a parallel printer.

The communication port test requires that a `serial port loopback plug` be installed in the port connector. This plug simulates another communication port connection. After the loopback plug is in place, this test transmits data through the port and then verifies that the received data is the same.

Using POST Cards

Objective

Most BIOS program chips do not have an extensive set of on-board diagnostics built into them. Therefore, several companies produce `POST Cards` and diagnostic software to aid in hardware troubleshooting. A POST card is a diagnostic device that plugs into the system's expansion slot and tests the operation of the system as it is booting up. These cards can be as simple as Interrupt and DMA channel monitors, or as complex as full-fledged ROM BIOS diagnostic packages that carry out extensive tests on the system.

POST cards are normally used when the system appears to be dead, or when the system cannot read from a floppy or hard drive. The firmware tests on the card replace the normal BIOS functions and send the system into a set of tests. The value of the card lies in the fact that the tests can be carried out without the system resorting to software diagnostics located on the hard disk or in a floppy drive.

The POST tests located in most BIOS chips stop the system if an error is detected. The POST cards note the error and continue through the initialization routine to activate as many additional system resources as possible.

Simple POST cards come with a set of LEDs (Light Emitting Diodes) on them that produce coded error signals when a problem is encountered. Other cards produce audio beep codes and 7-segment LED readouts of the error code. A typical XT/AT-compatible POST card is depicted in Figure 5.6.

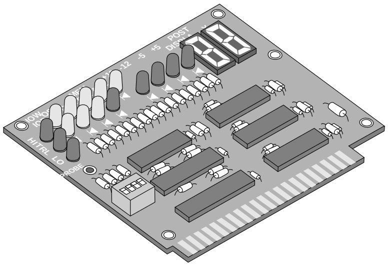


Figure 5.6

A typical POST card.

Using DOS Batch Files

Objectives

Many areas of the PC system can be tested using simple `DOS batch files`. It is not uncommon for technicians to create short files that test the monitor and floppy drive, as well as the serial and parallel ports. These files can be created through the `DOS Editor`, or by using the DOS `COPY CON` command. Any DOS command can be used to create a batch file.

FDDTest

A simple FDD test program can be constructed as follows:

```
Copy Con FDDTEST.BAT
DIR A:
FDDTEST.BAT
F6 <Enter>
```

This simple batch program creates a test program named FDDTEST. Typing this file name causes DOS to execute the two lines of the program. These lines cause the system to repeatedly perform directory listings of the floppy drive. The second line causes the program to rerun itself. Pressing the F6 function key stores the program to disk.

VIDTest

The DOS `echo` command can be used to set up a short test program for the video system. The program uses the echo command to display the message “This is a test of the video system” on the monitor screen. The second echo command, followed by a period, causes the program to create a blank line on the screen. The final line causes the program to automatically rerun and produce the message one line lower on the screen. In this manner, the screen should quickly fill up with the message:

```
Copy Con VIDTEST.BAT
Echo This is a test of the video system
Echo.
VIDTEST
F6 <Enter>
```

PRNTest

Similar test programs can be built to test I/O ports. DOS keeps track of the system’s installed ports by assigning them `handles` (logical device names) such as LPT1, LPT2, and LPT3. DOS also assigns COM port designations to the system’s serial ports during bootup. COM port designations are normally COM1 and COM2 in most systems, but they can be extended to COM4 in some advanced systems:

```
Copy Con PRNTEST.BAT
CD\DOS
PRINT lpt1 "This is a printer test
PRINT
PRNTEST
F6 <Enter>
```

The simple printer test batch program causes the system to change directories to the DOS subdirectory, execute a DOS print command to the first logical printer port, and print the message “This is a printer test” to the printer. The next command creates a blank line on the paper.

Batch Commands

These are all simple batch programs for very low-level testing of the system’s sub-sections. With the logical DOS and batch commands, you can create extensive diagnostic programs. Eight batch commands are particularly helpful in creating test files:

- Call
- Echo
- For
- Goto
- If
- Pause
- Rem
- Shift

Consult the DOS user's manual for more information and suggested usage of these commands.

Using Microsoft Diagnostics (MSD.EXE)

Microsoft Windows contains an excellent diagnostic program called `Microsoft Diagnostics` (MSD.EXE). It can enable you to determine what hardware options are installed in the computer without removing the case. It can also determine whether these options are responding to their correct addresses. MSD provides information about software packages installed during bootup (such as the operating system's version), which device drivers are installed, and what `Terminate and Stay Resident` programs (TSRs) are running. MSD is a good tool for eliminating or confirming software conflicts.

This software is particularly useful when adding new options to the system. As the system fills up with options, it becomes increasingly difficult to locate free system resources—such as DMA and interrupt channels—for the new options to use. By running MSD before installing a new option, you can avoid many hardware and software conflicts.

Although MSD is shipped with Windows, it is a DOS-based program. This means MSD does not have to be executed from within Windows. MSD should be executed from the DOS prompt by simply typing MSD at the C:\> prompt. The MSD's menu items can be selected using the mouse or the keyboard. When using the mouse, move the pointer over the desired menu selection and press the left mouse button. When using the keyboard, each menu selection has a highlighted letter that can be pressed to select the menu item.

When the MSD menu appears on the screen, you may choose from 13 options and a separate menu bar. They are as follows:

- Computer
- Memory
- Video
- Network
- OS Version
- Mouse
- Other Adapters
- Disk Drives
- LPT Ports
- COM Ports
- IRQ Status
- TSR Programs
- Device Drivers

The `MSD Toolbar` includes three options:

- File
- Utilities
- Help

The File pull-down menu enables the user to view system files, such as AUTOEXEC.BAT and CONFIG.SYS. It also shows Windows operating files, such as WIN.INI and SYSTEM.INI. Also under this menu is the option to search the disk drive(s) for a specific file. This option is called `File Find`. The Print a Report option, located in the File pull-down menu, produces a hard copy of items within each of the options mentioned in the preceding paragraphs, as well as any of the system file contents.

The Utilities pull-down menu enables the user to view memory allocation through the Memory Block Display option. Memory contents can also be viewed with the option Memory Browser. The menu also includes a Test Printer option, which can print an ASCII chart with the symbols and their decimal equivalents if a printer is connected.

The Help pull-down menu shows version and copyright information.

Using CheckIt

The `CheckIt` diagnostic program can be executed as a single module from the DOS command line to test a specific section of the system. It can also be run collectively through the GUI depicted in Figure 5.7. The program includes modules to test serial port(s), floppy-disk drive(s), hard-disk drive(s), parallel port(s), system board components, memory, and video operation. It also provides a virus-detection program as well as utilities to save/restore CMOS settings to/from disk, save device drivers to disk, collect configuration information, and calibrate a joystick. When any of the test application is executed, it produces a report that lists the results of the test.

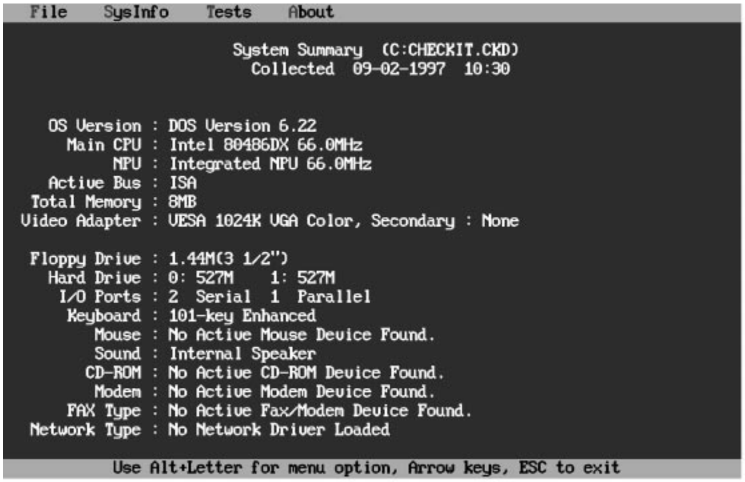


Figure 5.7

The CheckIt program screen.

One of the most interesting features of this program is its capability to customize the test procedure with the use of batch files. The program includes three of these batch files, which script three common test situations. The tests are called Quick test, Certification test, and Burn-in test. The tests vary in which sections of the system are tested and the number of iterations for the testing of each section.

An additional set of programs titled CheckIt Pro Sysinfo provides a wealth of information concerning the system's setup and performance. This information is particularly valuable when you are upgrading or installing a new option. The bank of programs enables you to evaluate the performance of the system before and after system changes so that maximum performance standards can be set.

Executing Command-Line Programs

The individual CheckIt programs are executed directly from the DOS command-line. The list in Table 5.2 describes the individual program modules included with CheckIt.

Table 5.2

CheckIt command line programs.

Program	Description
CKCMOS.EXE	Save and Restore CMOS Settings Utility
CKCOM.EXE	COM Port Test
CKDATA.EXE	Configuration Collection Utility
CKDRIVER.EXE	Save Device Driver Utility
CKFD.EXE	Floppy Drive Test
CKHD.EXE	Hard Drive Test
CKLPT.EXE	LPT Port Test
CKMEDIA.EXE	Data Integrity Test
CKMEM.EXE	Memory Test
CKRUN.EXE	Batch File Execution Utility
CKSYS.EXE	System Board Test
CKVID.EXE	Video Test

Executing the GUI Program

If multiple tests are to be executed, it may be easier to use the CheckIt programs collectively. This involves executing the GUI program included with the program. From the graphical interface, you can choose which tests should be run. Most parameters are specified before the test is actually executed. When the program starts, it enables you to collect data about that particular computer. It makes note of the microprocessor type, installed hardware, I/O address map, size and type of drives, and so forth. After the data has been collected, it can be saved and loaded into CheckIt the next time the program is executed on that computer.

The GUI main screen operates with `pull-down menus`. On the main screen is a summary of the system's information that was collected when the program started. The pull-down menus are manipulated by the keyboard. Use the ALT key (press and release) to highlight the menu bar, and the ARROW keys to select a single menu and menu option.

Using PC-Check

Another popular software diagnostic tool is `PC-Check`. PC-Check enables the technician to check system configuration and operation. It tests each part of the system and indicates those areas that do not respond correctly. Like other software diagnostic packages, PC-Check requires a working system core, a functional microprocessor, a functional floppy drive, and the display unit.

The types of tests run on the system are somewhat similar to those found in the CheckIt software described in the preceding section. PC-Check's Main Menu is depicted in Figure 5.8. The Main Menu is the gateway to information about the system's makeup and configuration as well as being the entryway to the PC-Check Advanced Diagnostic Test functions. Utilities for performing low-level formats on older hard drive types and managing SCSI interface devices are provided through this menu. Additionally, options to print or show test results are available here, as is the exit point from the program.

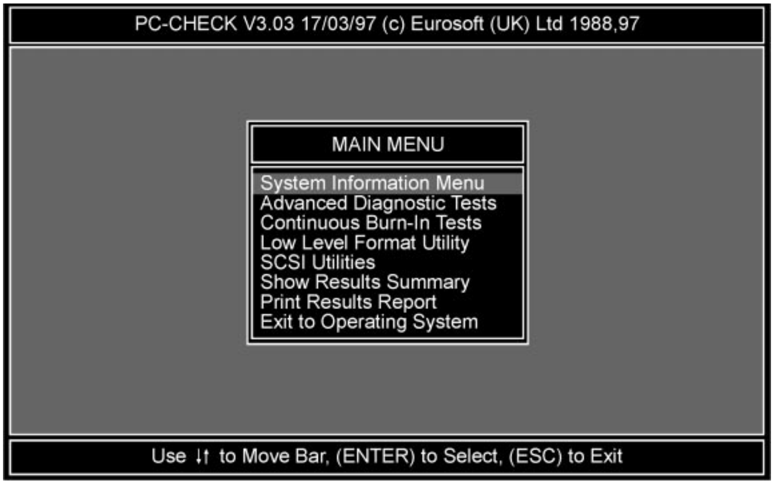


Figure 5.8

The PC-Check Main Menu.

The first option is the `System Information Menu`. This option provides access to the system's main functional blocks, as described in Figure 5.9. The menu's `IRQ Information`, `I/O Port Information`, and `Device Drivers` options are valuable aids in locating configuration conflicts.

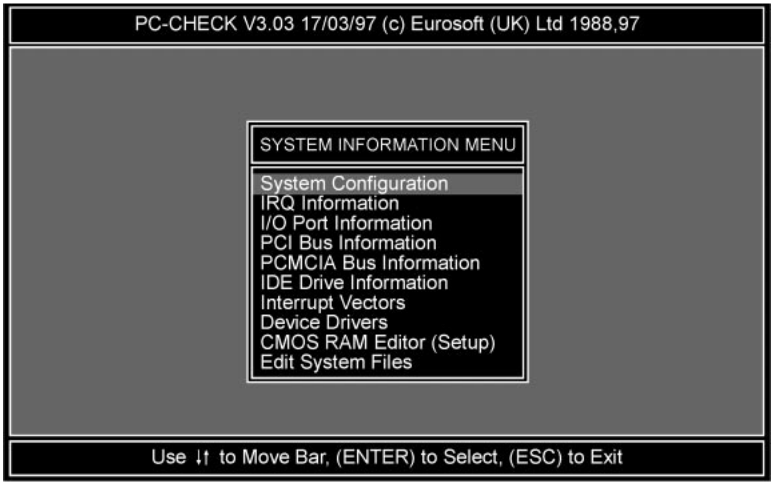


Figure 5.9

The PC-Check System Information Menu.

The `Advanced Diagnostics Tests` selection from the Main Menu performs extended tests in 13 system areas. (See Figure 5.10.) These tests contain several lower-level tests that can be selected from submenus. Error notices and diagnostic comments appear on the display in the form of overlay boxes.

Similar to the other software diagnostic packages, PC-Check performs a number of tests on the floppy drive (5), hard drive (6), input devices (4), display adapter (10), serial ports (1), and parallel ports (1). In addition to these fundamental software tests, PC-Check includes tests for multimedia-related devices such as CD-ROMs. The CD-ROM tests cover both access time and transfer performance. Both these values have an effect on the multimedia performance of the system. The multimedia tests also check the system's speaker and sound card capabilities.

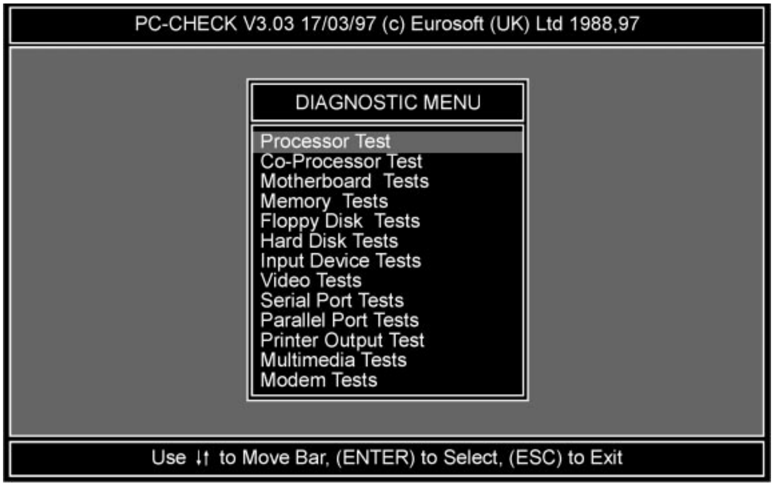


Figure 5.10

The PC-Check Diagnostic Menu.

For enterprises that repair computers, or build computers from parts, the PC-Check Continuous Burn In test is a valuable tool. After the system has been built or repaired, this function of the program runs continuous tests on the system for an extended (`Burn-In`) period of time without intervention from a technician or operator.

The tests performed are similar to the selection from the Main Menu. However, these tests are normally used for reliability testing rather than general troubleshooting. Different parts of the system can be selected for the burn-in tests. Because the burn-in tests are meant to be run unattended, the user must be careful to select only those tests that apply to hardware that actually exists. PC-Check keeps track of how many times each test has been run, how often it failed during the designated burn-in period. This information is displayed on the monitor.

Hardware Troubleshooting

Unfortunately, most software diagnostics packages do not lead you to specific components that have failed. Indeed, you may not even be able to use a software package to isolate faults if major components of the system are inoperative. If software and configuration problems have been eliminated, you need to pull out the test equipment and check the system's internal hardware for proper operation under controlled conditions.

Turn the power off and remove any peripheral devices from the system one at a time. Be sure to restore the power and retry the system after each peripheral is removed. If all the peripherals have been removed and the problem persists, it is necessary to troubleshoot the basic components of the system. This usually involves checking the components inside the system unit.

Performing Quick Tests

Objectives

After you've removed the cover of the system unit, perform a careful visual inspection of its interior. Look for signs of overheating, such as charred components or wires. When electronic components overheat, they produce a noticeable odor so you may be able to do some troubleshooting with your nose. If you do find an overheated component, especially a resistor, don't assume that the problem can be cleared up by simply replacing the burnt component. Many times when a component fails, it may cause another component to fail.

A very quick check of the system's integrated circuits can be made by simply touching the tops of the chips with your finger to see whether they're hot.

Note

> Because there may be MOS (Metal Oxide Semiconductor) devices on the board, you should ground yourself before performing this test. You can do this by touching an exposed portion of the unit's chassis, such as the top of the power supply.

If the system has power applied to it, all the ICs should be warm. Some are warmer than others by nature, but if a chip burns your finger, it's probably bad and needs to be replaced. But simply replacing the chip may not clear up your problem. Instead, you may end up with two dead chips; the original and the replacement. The original chip may have been wiped out by some other problem in the system. For this reason, this quick test should be used only to localize problems.

Other items to check include components and internal connections that may have come loose. Check for foreign objects that may have fallen through the device's air vents. Remove any dust buildup that may have accumulated, and then retry the system.

FRU Troubleshooting

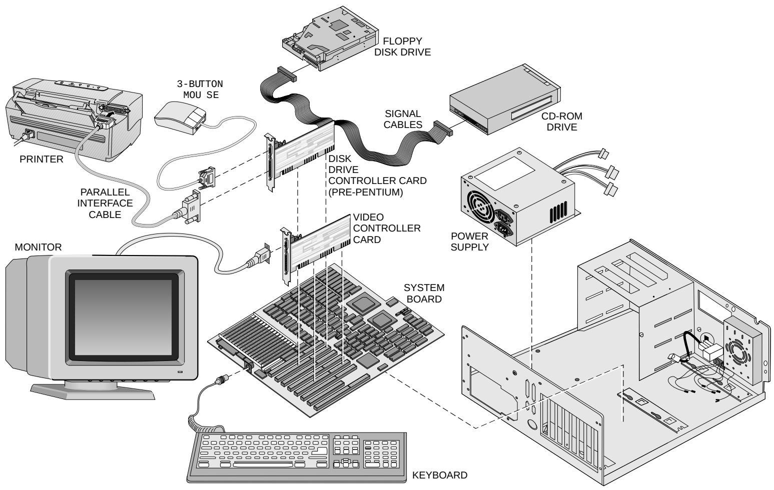
Objective

`Field Replaceable Units` (FRUs) are the portions of the system that can be conveniently replaced in the field. Typical microcomputer FRUs are depicted in Figure 5.11. FRU troubleshooting involves isolating a problem within one section of the system. A section consists of one device such as a keyboard, video display, video adapter card, I/O adapter card, system board, disk drive, printer, and so on. This is the level of troubleshooting most often performed on PCs. Due to the relatively low cost of computer components, it is normally not practical to troubleshoot failed components to the IC level. The cost of using a technician to diagnose the problem further, and repair it, can quickly exceed the cost of the new replacement unit.

After a hardware error has been indicated, start troubleshooting the problem by exchanging components (cards, drives, and so forth) with known good ones. When exchanging system components, be sure to replace the device being removed with one of exactly the same type. Just because two components have the same function `does not` mean that they can be substituted for each other. (For example, you cannot use an EGA video adapter card to replace a monochrome adapter card without making other modifications to the system.) Interchanging similar parts is possible in some cases and not in others. Whether or not two components can be exchanged depends on the particular modules.

Figure 5.11

The typical FRUs of a microcomputer system.



Assume that only a single component has failed. The odds against more than one component failing at the same time are extremely high. At the point where the system's operation is restored, it can be assumed that at least the last component removed was defective.

If it is necessary to disconnect cables or connectors from boards, take the time to mark the cables and their connection points, so that they will be easy to identify later. The simplest method of marking cables is to place identification marks on tape (masking or athletic) and then attach the tape to the cables and connection points.

Match the markings on the cable with the markings at its connection point. At many connection points, the color of the wire connected to a certain pin may be important. When placing the identifying marks on the tape, you may want to note the color arrangement of the wires being disconnected so that you can be sure of getting them back in their proper places after the component swap has been performed. Always check cabling connections after plugging them in. Look for missed connections, bent pins, and so forth.

It is often helpful to simply reseat (remove and reinstall) connections and adapter cards in the expansion slots when a problem occurs. Corrosion may build up on the computer's connection points and cause a poor electrical contact to occur. When you reseat the connection, the contact problem often disappears.

Note

> Make certain to take the time to document the symptoms associated with the problem, including all the tests you make, and any changes that occur during the tests. This information can keep you from making repetitive steps.

After you have isolated the problem, and the computer boots up and runs correctly, work backward through the troubleshooting routines, reinstalling any original boards and other components removed during the troubleshooting process. These steps should be performed one at a time, until all the original parts have been reinstalled except the last one removed. In this way, you can make certain that only one failure occurred in the machine. If the system fails after installing a new card, check the card's default configuration settings against the devices already installed in the system.

Special Troubleshooting Procedures

Objective

This section covers three important procedures: finding power supply problems, troubleshooting dead systems, and isolating undefined problems.

The power supply unit does not need to be developed from a conceptual point. Instead, it is treated as a simple passive FRU. The dead system and undefined problems procedures do not correspond to a single topic that can be expanded as described. Instead, these procedures can involve most of the components in the system.

Isolating Power Supply Problems

Typical symptoms associated with power supply failures include:

- No indicator lights visible, with no disk drive action, and no display on the screen. Nothing works; system is dead.

- The ON/OFF indicator lights are visible, but there is no disk drive action and no display on the monitor screen. The system fan may or may not run.
- A continuous beep tone is produced by the system.

The power supply unit is one of the few components in the system that is connected to virtually every other component in the system. Therefore, it has the capability to affect all the other components if it fails. Figure 5.12 illustrates the interconnections of the power supply unit with the other components in the system.

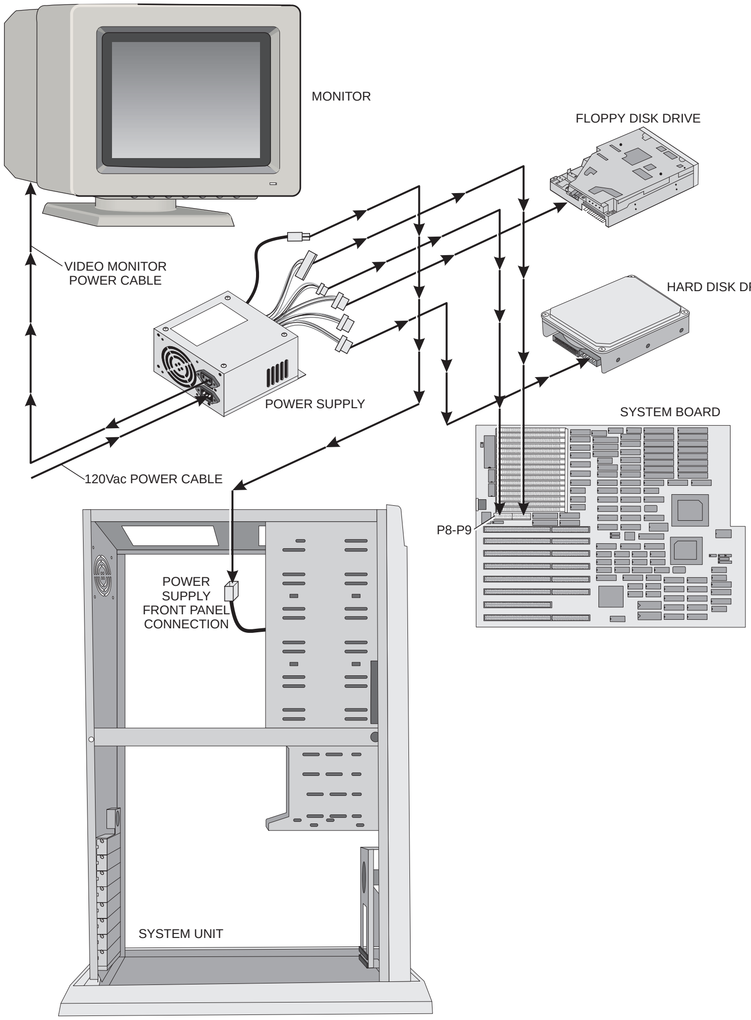


Figure 5.12

Power supply interconnections.

When you track down power supply problems, it's important to remember that in addition to the obvious power connections shown in the diagram, the power supply also delivers power to other components through the system board. These include: (1) all the options adapter cards (through the expansion-slot connectors) and (2) the keyboard (through the keyboard connector). Power supply problems can cause symptoms to occur in all these areas, and problems in any of these areas can affect the operation of the power supply.

Checking a Dead System

Objective

Special consideration must be taken when a system is inoperable. In a totally inoperable system, there are no symptoms to give clues where to begin the isolation process. In addition, it is impossible to use troubleshooting software or other system aids to help isolate the problem.

The following discussion is a standard method of troubleshooting dead microprocessor-based equipment. The first step in troubleshooting any dead system is to visually inspect the system. Check for unseated cards, loose cables, or foreign objects within the system unit.

When the system has no signs of life—including the presence of lights—the best place to start looking for the problem is the power supply. The operation of this unit affects virtually every part of the system. Also, the absence of any lights working usually indicates that no power is being supplied to the system by the power supply.

Begin by checking the external connections of the power supply. Confirm that the power supply cord is plugged into a functioning outlet. Check the position of the ON/OFF switch. Examine the power cord for good connection at the rear of the unit. Check the setting of the 110/220 switch setting on the outside of the power supply.

If power is reaching the power supply and nothing is happening, the next step in isolating the cause of the problem is to remove the peripheral devices so that only the basic system needs to be checked. Divide the system into basic and optional sections for testing. Remove all external options from the system and restart the system. If the system begins to work, troubleshoot the optional portions of the system.

Finally, divide the basic system into optional and basic components. Remove all optional adapter cards from their expansion slots and restart the system. If the system begins to work, troubleshoot the various options adapters by reinstalling them one at a time until the system fails again.

> **Note** Before changing any board or connection, always turn the system OFF first.

Isolating Undefined Problems

Normally, symptoms can be divided into three sections. These include Configuration Problems, Bootup Problems, and Operational Problems.

The system's configuration settings are normally checked first. It is important to observe the system's symptoms to determine in which part of the system's operation the fault occurs. Error messages typically associated with configuration problems include:

- CMOS Display Type Mismatch
- CMOS Memory Size Mismatch
- CMOS Battery State Low
- CMOS Checksum Failure
- CMOS System Options Not Set
- CMOS Time and Date Not Set

These errors occur and are reported before the single beep tone is produced at the end of the POST routines.

After the tone is produced, the system shifts over to the process of booting up. Typical error messages associated with bootup problems include:

- General Failure Error Reading Drive x
- Bad or Missing Command Interpreter
- Non-System Disk or Disk Error
- Bad File Allocation Table

Either type of problem can be caused by a hardware problem. If no configuration settings are incorrect but the symptoms are present, then a hardware problem is indicated. Likewise, bootup problems are typically associated with the operating system. However, hardware can also produce these symptoms.

Non-Classified Problems

Objectives

Some problems simply refuse to be classified under any particular symptom. If a multiple failure occurs, or if one failure causes a second failure to occur, the symptoms produced by the computer may not point to any particular cause. Secondary problems may also hide the symptoms of the real failure. It is best to simply begin with some logical starting point and work through the entire system until the problem is cured.

The system may be made up of the basic computer, monitor, and keyboard, or it may be a highly developed combination of equipment, involving the basic computer and a group of peripherals. For troubleshooting purposes, the system should be divided into logically related subsections.

The first division naturally falls between the components that make up the basic system and other devices. The basic system consists of the system unit, the keyboard, and the video display monitor. Other devices are the components which are optional as far as the system's operation is concerned, and which can be removed from the system without changing its basic operation. These items include such things as printers, mice, digitizing tablets, hard-disk drives, tape drives, scanners, and so forth.

Optional devices should be the first items removed from the system when a problem occurs. This divides the system in half and determines whether the problem exists in one of the computer's main components or in one of its options.

The second logical division falls between the internal and the external options. In cases where you have no idea of what the problem is, all external devices should be tested before removing the outer cover to check internal devices.

Inside the system unit, the next dividing point exists between the system board and all the internal options. The first items to be removed from the system are the options adapters, except for the disk drive and video controller cards. These cards should be checked only if the system still won't work properly with the other options adapters removed.

The next components to exchange are the floppy drives and the power supply unit, in that order. The system board is the last logical and most difficult component to exchange. Therefore, it should be the last component in the system to be exchanged.

Note

> The process for isolating non-classified problems is reinforced and expanded in the accompanying Hands-on Lab Book in Procedure 15.

Operating Systems Troubleshooting

An interesting troubleshooting point occurs at the single beep in the bootup process. If the system produces an error message, such as "The system has detected unstable RAM at location XXXX," or a beep-coded error signal before the beep, the problem is hardware-related. In this example, a bad memory device is indicated. Conversely, if the error message or beep code is produced after the beep, the problem is likely to be associated with the operating system.

Troubleshooting an operating system problem involves the same steps as any other logical troubleshooting procedure. The steps are simply adapted to fit the structure of the operating system. `Analyze` the symptoms displayed, `isolate` the error conditions, `correct` the problem, and `test` the repair.

When dealing with a disk operating system, three tools can be very useful in isolating the cause of operating system problems. These tools are system log files, clean boot disks, and single-step startup procedures. The preliminary steps involved in troubleshooting operating system problems are:

1. Try to reboot the system.
2. Check system log files if available.
3. Perform a clean boot with minimal configuration settings.
4. Perform a single-step bootup to isolate driver problems.

Troubleshooting DOS

MS-DOS problems can be divided into two basic categories: startup problems and operating problems. DOS does not produce a log file so it is necessary to perform a minimal bootup if the system does not respond to a normal reboot.

DOS Startup Problems

Typical DOS Startup Error messages include:

- Unrecognized command in CONFIG.SYS
- Bad or Missing Command interpreter

These and other error messages during the bootup process indicate that a problem exists that must be sorted out before the system can boot up and operate correctly. If the DOS system does not boot up correctly, it is necessary to use the F5 function key startup method to bypass the CONFIG.SYS and AUTOEXEC.BAT commands. This boots the system to a minimum configuration and establishes a point from which to begin troubleshooting the problem. The same function can be performed by running a clean boot disk to start the system. A guide for creating a clean boot disk is presented in the next section.

If the system boots up from the minimal condition, restart the system and press the F8 function key while the "Starting MS-DOS" message is on the screen to single-step through the CONFIG.SYS and AUTOEXEC.BAT files. The single-step method can be used to isolate the problem command. If the system crashes while trying to execute a particular command, restart the bootup process and skip the offending command. Repeat the process until the system reaches bootup. Track all offending commands so that they can be corrected individually.

Another common setup problem occurs when the system displays a "There is not enough free space on drive C to install MS-DOS" message. When this occurs, run the CHKDSK C:(check disk) command from the floppy drive. This provides a description of the free space available on the hard drive. Remove files from the disk until enough room has been cleared to perform the installation. It is recommended that the files be backed up to some other media before they are erased from the drive.

DOS Operating Problems

The most common DOS operating problems involve memory management issues. One of the most common errors occurs when a DOS program displays an "Out of memory" message. When this happens, it becomes necessary to free up additional memory. The first step in this process is to use the DOS MEM command to determine how much memory is actually in the system and how it is organized.

The real objective in memory management is to free up additional conventional memory. This can be accomplished by running the DOS Memmaker command from the DOS prompt. Memmaker moves device drivers and other memory resident programs into the system's upper memory area by modifying entries in the CONFIG.SYS and AUTOEXEC.BAT files. In particular, Memmaker changes switches associated with the EMM386.EXE entry. It also changes some of the device= lines to devicehigh= statements. It may also add loadhigh commands to some of the AUTOEXEC.BAT lines.

Use the steps outlined in the "Optimizing DOS" section that follows to free up as much memory as possible.

Another typical DOS operating error message is "Incorrect DOS version." This message is produced when a DOS utility, such as diskcopy, FDISK, MEM, and so forth, does not find the version of the operating system with which it is designed to work. This condition exists whenever a system has been booted from a different version of DOS than the one that resides on the hard drive, or when the system files of the hard disk have been repaired with a SYS command using a different version of DOS. In these cases, the correct version of DOS required to run the utility must be used. Use the **VER** command to determine which DOS version is in use.

Self-Booting DOS Disk

It is always good to have a clean boot disk to start the system. This tool provides a well-defined point from which to begin troubleshooting operating system problems. To create a self-booting DOS disk, place a new floppy disk in the floppy drive and enter the following lines at the DOS prompt:

```
Format A:/s
MD C:\DOS
CD \DOS
```

To make the disk truly useful, the following files should be copied to the boot disk, under the DOS directory:

```
FDISK
FORMAT
SYS
EDIT
CHKDSK
MSD
```

New minimum configuration CONFIG.SYS and AUTOEXEC.BAT files should be created for the startup disk. The files should include the following entries:

```
CONFIG.SYS
FILES=40
BUFFERS=40
SHELL=COMMAND.COM C:\DOS /p /e:256

AUTOEXEC.BAT
PATH=c:\;C:DOS
PROMPT $P$G
SET COMSPEC=C:\DOS
```

Optimizing DOS

The following steps can be used to optimize the operation of the system at the DOS level:

1. Use a **dos=high** or **dos=high,umb** command in the CONFIG.SYS file to load DOS into the HMA.

2. Check the CONFIG.SYS and AUTOEXEC.BAT files for lines that load the HIMEM.SYS, EMM386.EXE, SMARTDRV.EXE, and RAMDRIVE.SYS drivers. In each case, make certain that the latest version of the driver is located in the specified directory.

3. Check the order of commands in the CONFIG.SYS file to make certain that the HIMEM.SYS driver is loaded before any other extended memory application or driver. If not, move the command closer to the beginning of the file.

4. Set the memory cache size for the SMARTDRV.EXE command in the AUTOEXEC.BAT file to the largest size possible.

5. Optimize the CONFIG.SYS lines for buffers and files. Set files equal to 30 unless a currently installed application requires more handles. This step should also be used if DOS or Windows 3.x operations return a “Too many files are open” message. The number of buffers should be set to 10 if SMARTDRV is being used and 20 if not. Using more than 10 buffers with SMARTDRV decreases efficiency while using more than 20 buffers without SMARTDRV uses more of the system’s conventional memory area.

6. Set up the RAMDRIVE to use the TEMP environment. This improves printing performance and the operation of other applications that use .TMP files.

7. Load EMM386.EXE to allocate upper memory blocks for TSRs and device drivers.

Even when you use the previously listed setup steps, the system's performance deteriorates over time. Most of this deterioration is due to unnecessary file clutter and segmentation on the system's hard disk drive. The following steps can be used to periodically tune up the performance of the system. These steps are explained in greater detail in the Preventive Maintenance chapter.

1. Periodically remove unnecessary .TMP and .BAK files from the system.

2. Check for and remove lost file chains and clusters using the DOS CHKDSK and CHKDSK /f commands.

3. Use the DOS DEFRAG utility to realign files on the drive that may have become fragmented after being moved back and forth between the drive and the system.

Troubleshooting Windows 3.x

Until Windows 95, the Windows operating environment had a DOS layer running under it. Therefore, it was necessary to isolate DOS problems from Windows problems. Typical DOS problems were described in the preceding section. Typical Windows 3.x problems include:

- Windows setup problems
- Windows hardware problems
- Windows printing problems
- Windows operating problems
- Windows multimedia problems

This section examines only Windows 3.x problems associated with Setup and general Windows operation. Hardware, printing, and multimedia-related Windows problems are discussed in the chapters related to those topics.

Creating a Clean Boot Disk

Objectives

One of the best tools to have when troubleshooting Windows problems is a clean boot disk. When the Windows program becomes nonfunctional, it often becomes necessary to use the boot disk to restore the system to proper operation. The steps involved in creating the boot disk are:

1. Create a self-booting troubleshooting disk.
    a. Place a blank disk in drive A:.
    b. At the DOS prompt type **FORMAT A:/s** and press ENTER.
2. Create a new CONFIG.SYS file on the boot disk.
    a. At the C:\> DOS prompt, type: **COPY CON A:config.sys** and press ENTER.
    b. Enter the following lines of text, pressing the ENTER key at the end of each line:

```
buffers=20
files=40
stacks=9256
device=c:\windows\himem.sys
device=c:\windows\vga.sys
shell=c:\dos\command.com c:\dos /p /e:256
```

3. Save the new config.sys file to the floppy disk. Hold down the CONTROL key and press the Z key.
4. Create a new AUTOEXEC.BAT file on the boot disk.
    a. At the DOS prompt, type: **EDIT A:autoexec.bat**.
    b. Enter the following lines of text, pressing the ENTER key at the end of each line:

```
path=c:\dos;c:\windows
set temp=c:\windows\temp
prompt $p$g
```

5. Save the new AUTOEXEC.BAT file to the floppy disk.
   a. Hold down the ALT key and press the F key.
   b. Use the DOWN Arrow key to highlight the Save option and press ENTER.
6. Copy the system's .INI files to the floppy disk.
   a. copy \windows\win.ini a:
   b. copy \windows\system.ini a:
   c. copy \windows\progman.ini a:
   d. copy \windows\control.ini a:

The Clean Boot disk should be stored in a convenient space and labeled so that is easy to find. Without it, getting the system up and running after a crash becomes a much more difficult undertaking.

Windows 3.x Setup Problems

Two typical problems can occur when setting up Windows 3.x software. First, Windows can `hang up` (stop) the system during installation, or, second, the setup process can fail and return an error message to the screen.

Typical Windows 3.x setup error messages include:

- HIMEM.SYS not loaded
- Unable to initialize display adapter
- Swapfile corrupt
- Device referenced in WIN.INI could not be found

As with the DOS messages described earlier, these and other Windows 3.x startup error messages indicate that an error has occurred that must be remedied before the system can boot up and run properly. If the system hangs up while trying to run Setup, record any error messages displayed on the screen and troubleshoot that portion of the system. If no error message is returned, reboot the machine using the clean startup disk. Try to run Setup again.

Setup actually carries out two routines when it is executed. It first performs an MS-DOS setup routine to install Windows critical system files. This is followed by running a Windows Setup routine. The second routine sets up the Program Manager groups and Windows applications.

The Setup process has three common failure points. These occur primarily during the detection of TSRs, memory manager programs, and device drivers in the system. However, they can also occur during the auto-detection phase of the MS-DOS mode setup, as well as while Windows is being loaded into the system. In each case, it is possible that the system may do any of the three following:

Lock up totally

Give an error message

Simply return to the command line

If the system will not run the standard Setup routine after booting from the clean disk, attempt to run a modified Setup using the /t switch (that is, Setup /t). This switch setting checks the system for conflicting memory resident programs.

It is a good idea to check any TSR programs running in the system against the list of known problem TSRs located in the SETUP.TXT file on the Windows disk #1. Use the Windows SysEdit function to disable any TSRs in the system by REMing their statements in the AUTOEXEC.BAT and CONFIG.SYS files. This should make it possible to successfully install Windows after rebooting the system. The disabled TSRs can be added back to the system one at a time after Windows has been installed. When offending programs are found, try replacing them with standard Windows options available on the installation disks. If standard drivers do not work with the system, contact the manufacturer for Windows 3.x-compatible products.

The DOS Editor utility, or another word processor, can be used to examine the SETUP.INF (setup information) file in the System subdirectory for listings of problem TSRs. These files are listed under the [incompTSR1] and [incompTSR2] sections of the file.

Windows 3.x Operating Problems

Windows 3.x operation problems are of two general classes: General Protection Faults and Operating problems. Operating problems can be divided into Standard mode operating problems and 386 Enhanced mode operating problems.

General Protection (GP) Faults

A `General Protection` (GP) `Fault` occurs when Windows or one of its applications attempts to access an unallocated memory location. When this memory conflict occurs, the data in the violated memory location is corrupted and may crash the system. In Windows 3.0, a GP fault usually required that Windows be exited and the system be rebooted. Version 3.1 provided improved control of GP faults. In this version, the error notice includes information about where the error occurred and which application caused the error. In addition, Windows 3.1 remains stable enough after a GP fault to enable users to save the work in progress before exiting the program. Some GP faults are non-fatal and provide an option to ignore the fault and continue working, or to simply close the application. Although the application may continue to operate, it is generally not stable enough to continue working on an extended basis. It is recommended that the application be used only long enough to save any existing work.

Using Dr. Watson

Objectives

Windows 3.1 also includes a diagnostic tool, called Dr. Watson, that can be used to monitor the operation of the system and log its key events. The log provides a detailed listing of the events that led up to the failure. The information is automatically stored in a file named DRWATSON.LOG.

If Dr. Watson is not running, it should be activated so that it can serve as a reference for further troubleshooting steps. This can be accomplished by running the DRWATSON.EXE file from the Windows directory, or by installing its icon in the Startup window. This action causes Dr. Watson to run in the background each time Windows is started. Complete the following steps:

1. Check the DOS version to see that it is correct.
2. Check the HIMEM.SYS version to see that it is 3.01 or later.
3. Check the device drivers in the CONFIG.SYS file.
4. Perform a CHKDSK /F operation from the DOS prompt to check for cross-linked files.
5. If the fault occurs in 386 Enhanced mode, start Windows using the WIN /D:XSV switch.

   If the GP fault continues, it will be necessary to reinstall Windows from scratch.

   If the GP fault disappears, reduce the switch to WIN /D:XS.

   If the fault returns with the shortened switch, insert the line VirtualHDIRQ=OFF in the [386Enh] section of the SYSTEM.INI file.

   If the GP fault does not return, reduce the switch to WIN /D:X.

   If the fault returns, insert the line SystemPointBreakpoint=False in the [386Enh] section of the SYSTEM.INI file.

   If the GP fault does not return, search for an "EMMExclude=" line in the [386Enh] section of the SYSTEM.INI file. If the line is not present, create it, and set it to a value of A000-EFFF. If it is present, decrease the size of the address range.

Standard Mode Problems

If Windows has successfully been installed, changes in the system must occur to prevent Windows from running in Standard mode. These changes include:

- Low conventional memory
- Altered "PATH=" command
- Corrupted or missing files in the Windows directory
- Incorrect or missing HIMEM.SYS file
- Incorrect A20 handler routine being loaded
- Conflicting third-party device drivers
- Hardware devices installed after Windows was installed

If Windows will not start with the WIN /s switch, check to see that the machine has an 80286 or later microprocessor, with at least 256KB of conventional memory free. Also check to make certain that HIMEM.SYS, or some other extended memory manager, is installed with at least 192KB of extended memory free.

Attempt to boot the system using the clean boot disk and attempt to start Windows in Standard mode. If the system run Windows from the clean boot, then use the single-step bootup method to locate the offending configuration step so that it can be corrected.

Normally, if the system does not start in Standard mode, then some program or device is taking up too much memory. If the system has enough physical memory to run Windows, it is necessary to free up additional extended memory for use. Items to check include:

- Check the CONFIG.SYS and AUTOEXEC.BAT files for commands that load unnecessary programs into extended memory. Remove these commands and retry the system.

- Check the CONFIG.SYS file for memory managers and device drivers that use too much memory. Replace the driver with an updated driver or a standard Windows driver if possible.

- Check for a RAM drive installation that is using too much extended memory. Decrease the size of the RAM drive.

- The XMS driver is defective. The A20 handler routine of the HIMEM.SYS file may be incorrect. Check the HIMEM.SYS version to make sure that it is version 3.01 or later. Replace the HIMEM.SYS file with the version in the current Windows installation disks.

386 Enhanced Mode Problems

If Windows runs in Standard mode but will not run in 386 Enhanced mode, a conflict may exist in the allocation of the Upper Memory Blocks (UMBs). Windows may not detect the presence of an adapter card and try to use memory areas the card has taken if there are any of the following:

- An incorrect or missing HIMEM.SYS file

- An incorrect A20 handler routine being loaded

- Conflicting third-party device drivers

To detect this type of memory conflict, start Windows using a Win /d:x switch. This avoids the complete upper memory portion of the startup process. If Windows starts successfully using this switch, then a UMB conflict exists and needs to be corrected.

If conflicting device drivers are suspected, the DOS MEM /p command or the MSD.EXE diagnostic should be run to examine the drivers using the UMBs and their addresses. When a conflicting device driver has been located, add an “emmexclude=” line to the [386enh] section of the SYSTEM.INI file. The statement must include the address range of the device. The emmexclude statement prevents Windows from trying to use the space to establish buffers.

Shadow ROM and RAM should also be checked to see whether they are enabled in the E0000h area. If so, turn shadowing off for this memory section.

If the system crashes while running in enhanced mode, add the line **emmexclude=A000-EFFF** to the [386enh] section of the SYSTEM.INI file and restart Windows so that the change takes effect. This line prevents Windows from accessing the range of addresses between A000h and EFFFh.

If the system runs Windows enhanced mode after the SYSTEM.INI file has been modified, begin limiting the address range of the emmexclude command to find the location of the problem. It may also be necessary to turn off Shadow RAM features in these memory areas.

If the system produces a "Stack Overflow Error" message in Windows enhanced mode, check the "STACKS=" line in the CONFIG.SYS file. The stacks setting should be set to 9,256.

Reinstalling Windows 3.x

Many times, it is necessary to reinstall Windows 3.x to return a corrupted system to service. Usually, though, the technician should maintain the system's old settings if possible. To do this, it is necessary to install Windows in a new directory and try to run it from that location.

If the new installation does not run, use the DOS MEM command to check the system for low conventional memory. Also check for defective hardware and corrupted Windows files. If no apparent cause is present, completely erase the original Windows installation and reinstall Windows using standard Setup procedure.

If the new installation runs correctly, rename its .GRP and .INI files with .GRN and .INN (N = new) extensions. Copy all the files from the new installation into the old installation directories and try to run Windows. If the system still does not run, one of the old .INI file entries is probably corrupt. Rename the original .INI files and replace them with the .INN versions copied over from the new installation. As an alternative, reinstall all applications so that their drivers are updated throughout the system.

Optimizing Windows 3.x

The performance of a Windows 3.x system can be optimized by setting up the system using the following steps:

1. Use the 386 Enhanced icon in the Control Panel to establish a `Permanent Swap File` on the hard drive. (Select the fastest drive in multiple hard drive systems.) Also select the options in the Virtual Memory dialog box.

2. Select the lowest video resolution that is practical for the system. This option is selected through the Windows Setup icon or command. In most cases, this comes down to selecting the standard VGA driver from the option list.

3. Implement the DOS optimization steps given earlier in this chapter. Load DOS into the HMA as illustrated in the following CONFIG.SYS lines:

```
Device=path\HIMEM.SYS
Device=path\EMM386.EXE noems
DOS=HIGH,UMB
```

The `noems switch` in the EMM386 line establishes the upper memory blocks. A similar `ram switch` option provides UMBs, but imposes a 64KB EMS page frame. The noems switch does not produce a page frame and provides an additional 64KB in the UMB area. An `x= switch` can also be added to the noems line to exclude unused UMB addresses. This option enables Windows to move translation buffer memory that it establishes for non-Windows applications into the UMB. This frees up some additional conventional memory space. UMBs that can be excluded may be detected by using the MEM /p command at the DOS prompt.

Some optimization steps depend on the system's makeup and purpose. Generally, optimization considerations are divided between operating speed and disk space. The preceding steps all lead to optimizing the system for maximum speed. Additional speed-producing steps include:

- Remove any unnecessary TSRs and device drivers (such as non-Windows mouse drivers) from the AUTOEXEC.BAT and CONFIG.SYS files.
- Install SMARTDRV for caching.
- Create a permanent swap file for 386 Enhanced mode
- Establish the TEMP variable to the fastest hard disk in the system and the SwapDisk function to the drive with the most space available. This function is set up through the [Non-WindowsApp] section of the SYSTEM.INI file.

Conversely, to optimize a system for maximum hard disk space, perform the following steps:

1. Establish a RAMDRIVE using the TEMP environment.
2. Use a temporary swap file, or eliminate swapping altogether. This can be accomplished by entering "none" in the New Settings Type box of the Control Panel's Virtual Memory dialog box.
3. Use the minimum Windows installation possible.

Troubleshooting Windows 95

Windows 95 offers many improved features over previous operating systems; however, it can suffer many of the same problems as any other operating system. However, Windows 95 includes several built-in troubleshooting tools. These tools include several Safe Mode startup options, a trio of system log files, and an extensive interactive troubleshooting help file system.

Windows 95 Startup Problems

Windows 95 requires 420KB of conventional memory to start. It requires at least 3MB of extended memory as well. As with previous operating systems, two important tools can be used with a system that is having startup problems. These are the clean boot disk and the step-by-step startup sequence. In the case of Windows

95, the clean disk is referred to as an `Emergency Start Disk`. The single-step startup process can be accessed by pressing the Shift and F8 function keys simultaneously when the “Starting Windows 95” message appears on the screen.

Typical Windows 95 Startup Error messages include:

- HIMEM.SYS not loaded
- Unable to initialize display adapter
- Device referenced in WIN.INI could not be found
- Bad or missing COMMAND.COM
- Swapfile Corrupt

These and other Windows 95-related startup messages indicate the presence of problems that must be corrected before the system can boot up and run correctly. As with DOS and Windows 3.x startup problems, the logical procedure for isolating and correcting these problems in Windows 95 involve booting the system from the emergency start disk, or starting the system in safe mode, and then single-stepping through the startup sequence until the offending steps have been isolated.

Creating an Emergency Start Disk

Objectives

Because Windows 95 does not start up through DOS, it is very difficult to gain access to the system if Windows becomes disabled. Therefore, it is helpful to have a clean Startup disk to troubleshoot Windows 95-related problems. If the Windows program ever becomes nonfunctional, it will be necessary to use the Startup disk to restore the system to proper operation.

In addition to creating a Startup floppy, Windows 95 transfers a number of diagnostic files to the disk. These utilities are particularly helpful in getting a Windows 95 machine operational again. Because the only path to DOS is through Windows, this disk provides one of the few tools for the technician to service a down machine with this operating system. The steps involved in creating the boot disk are:

1. Create the Emergency Startup Disk.
   a. Click on the Start button.
   b. Move to the Settings option in the Start menu.
   c. Select the Control Panel from the list.
   d. Double-click on the Add/Remove Programs icon.
   e. Select the Startup Disk tab.
   f. Click on the Create Disk button.
   g. Place the WIN 95 CD in the CD-ROM drive when prompted.
   h. Follow the menu items as directed.
   i. Place a blank floppy disk in Drive A when prompted.
   j. Remove the WIN 95 CD from the drive when the operation is complete.
2. Examine the Startup disk.
   a. Close the Control Panel window.
   b. Select the Windows Explorer option from the Start/Programs menu.
   c. Click on the 3.5" Floppy A: option from the list in the All Folders window.
   d. Label the disk as an Emergency Startup Disk.
3. Examine the new CONFIG.SYS file on the boot disk.
   a. Select the Notepad utility in the Start/Programs/Accessories menu.
   b. Click on the File and Open options in the Notepad window.

c. Select the 3.5" Floppy (A:) option in the Look In window.

d. Select the All File (*.*) option from the Files of Type window.

e. Double-click on the CONFIG.SYS entry in the window.

4. Add helpful files to the Start disk.

a. Close the Notepad utility.

b. Select the Windows Explorer option from the Start/Programs menu.

c. Click on the File and Open options in the Notepad window.

d. Select the 3.5" Floppy (A:) option in the Look In window.

e. Select the Folder option from the File/New menu.

f. Type your three initials in the box for the new subdirectory.

g. Select the (C:) option in the Look In window.

h. Locate the AUTOEXEC.BAT file in the "Contents of C:\" window.

i. Click, hold, and drag the file to the 3.5" Floppy (A:) option on the Look In window, then release.

j. Repeat Step 5-i for the SYSTEM.DAT, CONFIG.SYS, WIN.INI, and SYSTEM.INI files.

k. Exit the Windows Explorer.

5. Examine the new AUTOEXEC.BAT file on the boot disk.

a. Select the Notepad utility in the Start/Programs/Accessories menu.

b. Click on the File and Open options in the Notepad window.

c. Select the 3-1/2" Floppy (A:) option in the Look In window.

d. Select the All File (*.*) option from the Files of Type window.

e. Double-click on the AUTOEXEC.BAT entry in the window.

As with the DOS and Windows 3.x Boot Disks, the Windows 95 Emergency Start Disk should be stored in a convenient place and clearly labeled so that it is easy to find when you need it.

WIN Switches

When Windows 95 refuses to start up, a number of options are available for starting it from the command line. Starting Windows using a /d switch is often helpful in this situation (for example, WIN /d). The /d switch can be modified to start Windows in a number of different configurations.

Using an `:f modifier` disables 32-bit disk access. The `:m` and `:n modifiers` start Windows in Safe mode, or Safe with Networking mode. An `:s modifier` inhibits Windows from using address space between F0000h and FFFFFh. The `:v modifier` prevents Windows from controlling disk transfers. Instead, HDD interrupt requests are handled by the BIOS. Using the `:x modifier` prevents Windows from using the A000h-FFFFh area of memory.

Windows 95 Log Files

Windows 95 maintains four log files named BOOTLOG.TXT, SETUPLOG.TXT, DETLOG.TXT, and DETCRASH.LOG. These files maintain a log of different system operations and can be used to see what events occurred leading up to a failure. The .TXT files can be read with Notepad, DOS Editor, or any other text editor.

BOOTLOG.TXT

The BOOTLOG.TXT file contains the sequence of events conducted during the startup of the system. A bootlog can be created by pressing the F8 key during Startup, or by starting Windows 95 with a **WIN /b** switch. The log information is recorded in five basic sections.

- **Loading Real Mode Drivers**: This section records a two-part loading report during the bootup process. In the example section that follows, the system successfully loads the HIMEM.SYS and EMM386.EXE memory managers. Afterward, a list of other Real mode drivers is loaded. In the case of an unsuccessful load operation, the report returns a **LoadFailed=** entry.

```
[000E3FDC] Loading Device = C:\WINDOWS\HIMEM.SYS
[000E3FE0] LoadSuccess    = C:\WINDOWS\HIMEM.SYS
[000E3FE0] Loading Device = C:\WINDOWS\EMM386.EXE
[000E3FEC] LoadSuccess    = C:\WINDOWS\EMM386.EXE
[000E3FEC] Loading Device = C:\WINDOWS\SETVER.EXE
[000E3FEF] LoadSuccess    = C:\WINDOWS\SETVER.EXE
[000E3FEF] Loading Device = C:\PWRSCSI!\MCAM950.SYS
[000E406B] LoadSuccess    = C:\PWRSCSI!\MCAM950.SYS
[000E406B] Loading Device = C:\PWRSCSI!\FDCD.SYS
[000E40BD] LoadSuccess    = C:\PWRSCSI!\FDCD.SYS
[000E40BD] Loading Device = C:\WINDOWS\IFSHLP.SYS
[000E40C0] LoadSuccess    = C:\WINDOWS\IFSHLP.SYS
```

- **Loading VxDs**: In the second section, the system loads the VxD drivers. The following list includes a sample of various VxDs that have been loaded. The asterisks in the sample listing are included to indicate sections of omitted lines. This is done to shorten the length of the file for illustration purposes.

```
[000E4198] Loading Vxd = VMM
[000E419B] LoadSuccess = VMM
*
[000E41AA] Loading Vxd = CONFIGMG
[000E41AB] LoadSuccess = CONFIGMG
*
[000E41AC] Loading Vxd = VWIN32
```

```
[000E41AC] LoadSuccess = VWIN32
[000E41AC] Loading Vxd = VFBACKUP
[000E41AC] LoadSuccess = VFBACKUP
[000E41AC] Loading Vxd = VCOMM
[000E41AD] LoadSuccess = VCOMM
[000E41AD] Loading Vxd = COMBUFF
[000E41AD] LoadSuccess = COMBUFF
*
[000E41B1] Loading Vxd = VFAT
[000E41B2] LoadSuccess = VFAT
[000E41B2] Loading Vxd = VCACHE
[000E41B2] LoadSuccess = VCACHE
*
[000E41B5] Loading Vxd = V86MMGR
[000E41B6] LoadSuccess = V86MMGR
[000E41B6] Loading Vxd = PAGESWAP
[000E41B6] LoadSuccess = PAGESWAP
*
[000E41F3] Loading Vxd = int13
[000E41F3] LoadSuccess = int13
[000E41F3] Loading Vxd = vmouse
[000E41F4] LoadSuccess = vmouse
[000E41F6] Loading Vxd = msmouse.vxd
[000E41F9] LoadSuccess = msmouse.vxd
[000E41F9] Loading Vxd = vshare
[000E41F9] LoadFailed  = vshare
*
[000E4208] Loading Vxd = EBIOS
[000E4208] LoadFailed  = EBIOS
```

- **Initialization of Critical VxDs**: Check this section to verify that system-critical VxDs have been initialized.

```
[000E420A] SYSCRITINIT  = VMM
[000E420A] SYSCRITINITSUCCESS  = VMM
[000E420A] SYSCRITINIT  = VCACHE
[000E420A] SYSCRITINITSUCCESS  = VCACHE
[000E420A] SYSCRITINIT  = PERF
[000E420A] SYSCRITINITSUCCESS  = PERF
[000E420A] SYSCRITINIT  = VPICD
[000E420A] SYSCRITINITSUCCESS  = VPICD
[000E420A] SYSCRITINIT  = VTD
[000E420A] SYSCRITINITSUCCESS  = VTD
*
*
```

- **Device Initialization of VxDs**: This section of the log shows the VxDs that have been successfully initialized. In each cycle, the system attempts to initialize a VxD and then reports its success or failure.

```
[000E420D] DEVICEINIT   = VMM
[000E420D] DEVICEINITSUCCESS   = VMM
[000E420D] DEVICEINIT   = VCACHE
[000E420D] DEVICEINITSUCCESS   = VCACHE
[000E420E] DEVICEINIT   = PERF
[000E420E] DEVICEINITSUCCESS   = PERF
*
[000E421F] Dynamic load device  isapnp.vxd
[000E4225] Dynamic init device  ISAPNP
[000E4226] Dynamic init success ISAPNP
[000E4226] Dynamic load success isapnp.vxd
*
*
```

The bold information in the listing points out the dynamic loading and initialization of the ISA bus PnP driver.

- **Successful Initialization of VxDs**: The entries in this section verify the successful completion of the initialization of the system's VxDs. A partial listing of these activities follows:

```
[000E4430] INITCOMPLETE = VMM
[000E4430] INITCOMPLETESUCCESS = VMM
[000E4430] INITCOMPLETE = VCACHE
[000E4430] INITCOMPLETESUCCESS = VCACHE
[000E4430] INITCOMPLETE = PERF
[000E4430] INITCOMPLETESUCCESS = PERF
*
*
```

SETUPLOG.TXT

The SETUPLOG.TXT file holds setup information that was established during the installation process. The file is stored on the system's root directory and is used in Safe Recovery situations. The log file exists in seven basic sections, as described in the following sample sections. Entries are added to the file as they occur in the setup process. Therefore, the file can be read to determine what action was being taken when a setup failure occurred.

```
[OptionalComponents]
"Accessories"=1
"Communications"=1
"Disk Tools"=1
"Multimedia"=1
"Screen Savers"=0
"Disk compression tools"=1
"Paint"=1
"HyperTerminal"=1
"Defrag"=1
"Calculator"=1
"Backup"=0
"Phone Dialer"=1
"Flying Windows"=1
"Desktop Wallpaper"=0
*
*
[System]
"Display"="S3"
"Keyboard"="Standard 101/102-Key or Microsoft Natural Keyboard"
"SelectedKeyboard"="KEYBOARD_00000409"
"MultiLanguage"="ENGLISH"
"Machine"="MS_CHICAGO"
"Monitor"="(Unknown Monitor)"
"Mouse"="Standard Serial Mouse"
"Power"="No_APM"
"Locale"="L0409"
"UI_Choice"="Win95UI"
*
*
[Setup]
InstallType=1
Customise=0
Express=0
ChangeDir=1
Network=1
OptionalComponents=1
System=1
MBR=1
Reboot=1
*
CleanBoot=0
Win31FileSystem=-8531
CopyFiles=1
```

```
Verify=-8531
*
CHKDSK=0
UNINSTALL=1
DevicePath=0
InstallDir=C:\WINDOWS,0,400
[Started]
version=262144,950
OldLogFile
SourcePath=C:\WININST0.400
CmdLine=/T:C:\WININST0.400 /SrcDir=D:\WIN95 /IZ /IS /IQ /IT /II /
➥C  /U:xxxxxxxxxxxxxxxxx
WinVer=Mini
ExePath=C:\WININST0.400
FilePath=D:\WIN95\
RunVer=3.1
Init:Setup initialization sucessful.
Started=Passed

[Detection]
Detection=Passed
Detection Passed
loading 'C:\WININST0.400\suexpand.dll' returned c4f0ea6
VcpClose:About to close
VcpClose:About to End
VcpClose:About to Terminate
CacheFile() C:\WINDOWS\system.ini returns=0
*
Display_InitDevice:Checking display driver. No PNP registry
➥found.
Mouse_InitDevice:Checking mouse driver. No PNP registry found.
```

The bold text in the sample section shows the system's response to not finding a PnP registry entry for different devices.

```
[FileQueue]
CacheFile() C:\WINDOWS\win.ini returns=0
CacheFile() C:\WINDOWS\exchng32.ini returns=0
CacheFile() C:\WINDOWS\control.ini returns=0
CacheFile() C:\WINDOWS\qtw.ini returns=0
CacheFile() C:\WINDOWS\system.cb returns=0
SrcLdid:(11)skyeagle.cpl
CacheFile() C:\WINDOWS\msoffice.ini returns=0
SrcLdid:(25)reg.dat
```

```
Winver (0400) specific Section:win4.0, reRet=0.
Calling Netdi to install
CacheFile() C:\WINDOWS\protocol.ini returns=0
*
*
[FileCopy]
VcpClose:About to close
VcpClose:Delete 1514
VcpClose:Rename 4
VcpClose:Copy 827
CAB-No volume name for LDID 2, local copy - path C:\WININST0.400
adapter.inf=17,,7915,20032
appletpp.inf=17,,7915,20032
applets.inf=17,,7915,20032
awfax.inf=17,,7915,20032
awupd.inf=17,,7915,20032
bkupagnt.inf=17,,7915,20032
cemmf.inf=17,,7915,20032
cheyenne.inf=17,,7915,20032
command.com=30,command.new,7915,20032
decpsmw4.inf=17,,7915,20032
diskdrv.inf=17,,7915,20032
drvspace.bin=13,,7915,20032
drvspace.sys=13,,7915,20032
dskmaint.dll=11,,9063,24692
enable.inf=17,,7915,20032
extract.exe=13,,9063,24685
fonts.inf=17,,7915,20032
*
*
[Restart]
loading 'C:\WININST0.400\suexpand.dll' returned c4f0ea6
SrcLdid:(30)SYSTEM.NEW
SrcLdid:(30)COMMAND.NEW
SrcLdid:(30)WINBOOT.NEW
SrcLdid:(26)SETVER.WIN
SrcLdid:(30)SULOGO.SYS
SrcLdid:(30)USER.NEW
Ver:C:\DRVSPACE.BIN=:262144:256:
➥Ver:C:\WININST0.400\DRVSPACE.BIN=:262144:256:
SrcLdid:(2)DRVSPACE.BIN
Ver:C:\DBLSPACE.BIN=:262144:256:
➥Ver:C:\WININST0.400\DRVSPACE.BIN=:262144:256:
SrcLdid:(2)DRVSPACE.BIN
```

```
VcpClose:About to close
VcpClose:Delete 101
VcpClose:Copy 10
CAB-No volume name for LDID 2, local copy - path C:\WININST0.400
Resolve Conflict:C:\DBLSPACE.BIN ConflictType: 240
drvspace.bin=31,DBLSPACE.BIN,7915,20032
Resolve Conflict:C:\drvspace.bin ConflictType: 240
drvspace.bin=31,,7915,20032
drvspace.sys=13,dblspace.sys,7915,20032
LDID is ffff failed CtlGetLdd
CAB-No volume name for LDID ffff, local copy - path Absolute
COMMAND.NEW=30,command.com,7915,20032
COMMAND.NEW=26,command.com,7915,20032
*
*
```

The bold lines in the example demonstrate the capability of the PnP system to resolve conflicts between programs and devices. In this case, there is a conflict between a driver named DBLSPACE and another named DRVSPACE.

DETCRASH.LOG

This `Detect Crash` log file is created when the system crashes during the hardware detection portion of the startup procedure. It contains information about the detection module that was running when the crash occurred. This file is a binary file and cannot be read directly. However, a text version of the file, named DETLOG.TXT is available under the root directory of the drive.

DETLOG.TXT

The DETLOG.TXT file holds the text equivalent of the information in the DETCRASH.LOG file. This file can be read with a text editor to determine which hardware components have been detected by the system and what its parameters are. The following section of a sample DETLOG.TXT file demonstrates the type of information logged in this file.

```
[System Detection: 11/07/97 - 12:05:11]
Parameters "", InfParams "", Flags=01004233
SDMVer=0400.950, WinVer=0700030a, Build=00.00.0,
➥WinFlags=00000419
```

```
LogCrash: crash log not found or invalid
SetVar: CDROM_Any=
Checking for: Programmable Interrupt Controller
QueryIOMem: Caller=DETECTPIC, rcQuery=0
     IO=20-21,a0-a1
Detected: *PNP0000\0000 = [1] Programmable interrupt controller
     IO=20-21,a0-a1
     IRQ=2
Checking for: Direct Memory Access Controller
QueryIOMem: Caller=DETECTDMA, rcQuery=0
     IO=0-f,81-83,87-87,89-8b,8f-8f,c0-df
Detected: *PNP0200\0000 = [2] Direct memory access controller
     IO=0-f,81-83,87-87,89-8b,8f-8f,c0-df
     DMA=4
Checking for: System CMOS/Real Time Clock
QueryIOMem: Caller=DETECTCMOS, rcQuery=0
     IO=70-71
Detected: *PNP0B00\0000 = [3] System CMOS/real time clock
     IO=70-71
     IRQ=8
Checking for: System Timer
QueryIOMem: Caller=DETECTTIMER, rcQuery=0
     IO=40-43
Detected: *PNP0100\0000 = [4] System timer
     IO=40-43
     IRQ=0
Checking for: System Speaker
QueryIOMem: Caller=DETECTSPEAKER, rcQuery=0
     IO=61-61
Detected: *PNP0800\0000 = [5] System speaker
     IO=61-61
Checking for: Numeric Data Processor
QueryIOMem: Caller=DETECTNDP, rcQuery=0
     IO=f0-ff
Detected: *PNP0C04\0000 = [6] Numeric data processor
     IO=f0-ff
     IRQ=13
Checking for: System Board
Detected: *PNP0C01\0000 = [7] System board
Checking for: Keyboard
QueryIOMem: Caller=DETECTKBD, rcQuery=0
     IO=60-60,64-64
GetKbdType: Keyboard ID=faab41
Detected: *PNP0303\0000 = [8] Standard 101/102-Key or Microsoft
```

```
➥Natural Keyboard
     IO=60-60,64-64
     IRQ=1
Checking for: System Bus
CheckInt86xCrash: int 1a,AX=b101,rc=0
Detected: *PNP0A00\0000 = [9] ISA Plug and Play bus
Checking for: S3 801/805/928/864/964 Display Adapter
QueryIOMem: Caller=DETECTS3801, rcQuery=0
     Mem=c0000-c07ff
QueryIOMem: Caller=DETECTS3801, rcQuery=0
     Mem=c0000-c7fff
VerifyHWReg: failed verification of *PNP0913\0000
Checking for: Future Domain TMC-950 SIC-based SCSI Host Adapter
QueryIOMem: Caller=DETECTFD8XX, rcQuery=0
     Mem=cbc00-cbfff
Detected: *FDC0950\0000 = [10] Future Domain TMC-850/M/MER/MEX
➥SCSI Host Adapter
     Mem=ca000-cbfff
     IRQ=5
Checking for: Standard Floppy Controller
QueryIOMem: Caller=DETECTFLOPPY, rcQuery=0
     IO=3f2-3f5
QueryIOMem: Caller=DETECTFLOPPY, rcQuery=0
     IO=372-375
Detected: *PNP0700\0000 = [11] Standard Floppy Disk Controller
     IO=3f2-3f5
     IRQ=6
     DMA=2
Checking for: Serial Communication Port
QueryIOMem: Caller=DETECTCOM, rcQuery=0
     IO=3f8-3ff
GetCOMIRQ: IIR=1
Detected: *PNP0500\0000 = [12] Communications Port
     IO=3f8-3ff
     IRQ=4
SetVar: COMIRQ3f8=4,0
Checking for: Serial Communication Port
QueryIOMem: Caller=DETECTCOM, rcQuery=0
     IO=2f8-2ff
GetCOMIRQ: IIR=1
Detected: *PNP0500\0001 = [13] Communications Port
     IO=2f8-2ff
     IRQ=3
SetVar: COMIRQ2f8=3,0
Checking for: Serial Mouse
```

```
QueryIOMem: Caller=DETECTSERIALMOUSE, rcQuery=2
     IO=3f8-3ff
Serial mouse ID: M (004d)
Detected: *PNP0F0C\0000 = [14] Standard Serial Mouse
SetVar: COMIRQ3f8=4,0
Checking for: Standard IDE/ESDI Hard Disk Controller
QueryIOMem: Caller=DETECTESDI, rcQuery=0
     IO=1f0-1f7
QueryIOMem: Caller=DETECTESDI, rcQuery=0
     IO=3f6-3f6
Detected: *PNP0600\0000 = [15] Standard IDE/ESDI Hard Disk
➥Controller
     IO=1f0-1f7,3f6-3f6
     IRQ=14
Checking for: Parallel Printer Port
QueryIOMem: Caller=DETECTLPT, rcQuery=0
     IO=378-37a
Detected: *PNP0400\0000 = [16] Printer Port
     IO=378-37a
Checking for: Sound Blaster Compatible
SetVar: SkipGUSResources=
QueryIOMem: Caller=DETECTSB, rcQuery=0
     IO=220-22f
QueryIOMem: Caller=DETECTSB, rcQuery=0
     IO=388-38b
Detected: *ESS4881\0000 = [17] ESS ES488 AudioDrive
     IO=220-22f,388-389
     IRQ=7
     DMA=1
Checking for: Sound Blaster Compatible
QueryIOMem: Caller=DETECTSB, rcQuery=0
     IO=201-201
Detected: *PNPB02F\0000 = [18] Gameport Joystick
     IO=201-201
VerifyHW: manual device UNIMODEM535147EF\COM2: 14.4 Data FAX
➥Modem
```

Referring to the information in the sample file, it should be easy to see the type of information that is logged about the system. The detection routine cycles through a three-part process: First it identifies the activity it is about to perform (for example, Checking for: Serial Mouse), then it Queries the system at addresses normally allocated to that type of device (IO=3f8-eff), and finally verifies that it was Detected (or not). Some entries also include a listing of the IRQ and DMA resources allocated to the device. The sample list includes information about many of the system and I/O devices already mentioned in Chapters 1 and 2.

In each case, the Plug-and-Play (PnP) system inquires about particular system devices and logs the parameters of the device it detects. The sample also shows that, at least in this case, no crash log has been created. To use the file for crash detection purposes, simply check the last entry created in the log. To determine exactly where a problem has occurred, it may be necessary to compare this information to the listing in a file named DETLOG.OLD. This file is the old version of the DETLOG file that was renamed before the latest detection phase began.

Windows 95 Operating Problems

The Windows 95 structure provides a much improved multitasking environment over Windows 3.x. However, applications can still attempt to access unallocated memory locations, or attempt to use another application's space. When these memory conflicts occur, the system can either return an error message, or simply stop processing.

Windows 95 typically indicates that such a problem has occurred by placing a "This program has committed an Illegal Operation and is about to be shut down" message on the screen. When this happens, Windows may take care of the error and allow you to continue operating by simply pressing a key.

If the system locks up, it is possible to gain access to the Windows task list by pressing the CTRL/ALT/DEL key combination. Once the task list is on the screen it is possible to close down the offending application and continue operating the system without rebooting.

If a DOS-based program was running and the system locked up, it will be necessary to restore Windows 95. To accomplish this, try to restart the system from a cold boot. If the system comes up in Windows 95, check the properties of the DOS application under the Advanced button of the My Computer/filename/Properties/ Program path. If the application is not already set for MS-DOS mode operation, click the box to select it. Also select the "Prevent MS-DOS-based programs from detecting Windows" option. Return to the failing application to see if it will run correctly in this environment.

If the DOS program causes the system to stall during Windows 95 bootup, shut down the system and then bootup the computer. When the "Starting Windows 95" message appears on the screen press the F8 key. This will bring up the Startup menu. Select the "Restart in MS-DOS mode" option from the list. From this point, it will be necessary to edit the AUTOEXEC.BAT and CONFIG.SYS files to remove selected lines. In the AUTOEXEC.BAT file **rem** the following lines:

- \Windows\command
- \call c:\windows\command***
- \windows\win.com/wx

Also **rem** the "dos=single" line in the CONFIG.SYS file.

Windows 95 Help Files

Windows 95 comes with a built-in Troubleshooting Help file system. This feature includes troubleshooting assistance for a number of different Windows 95 problems. It can be accessed from the Start button, or from the Help menu entry on the task bar. In either case, the Help Topic window appears as described in Figure 5.13.

Double-clicking on the Troubleshooting entry accesses the Troubleshooting Help section. This section contains a list of several entries with information about common Windows problems and situations. Clicking on a topic produces a Help window with information about the troubleshooting process associated with that particular problem (for example, Hardware Conflict Troubleshooting). This window is depicted in Figure 5.14. The interactive text contains a step-by-step procedure for isolating the problem listed.

Figure 5.13

The Windows Help Topics window.

Using Device Manager

Hardware and configuration conflicts can also be isolated manually by using the Windows 95 `Device Manager` from the Control Panel's System icon. The Device Manager can be used to identify installed ports, update device drivers, and change I/O settings. The Device Manager will display an exclamation point (!) inside a yellow circle whenever a device is experiencing a direct hardware conflict with another device. Similarly, when a red X appears at the device's icon, the device has been disabled to a User Selection Conflict. When a conflict is suspected, click on the offending device in the listing, make sure that the selected device is the current device, and then click the Resources tab to examine the `Conflicting Devices list`. Make sure that the device has not been installed twice.

The System Tools

Windows 95 includes several helpful utilities that can be accessed through the Start/Programs/Accessories/System Tools path. Among these utilities are the `System Monitor` and `System Resource Meter` programs. The System Monitor can be used to track the performance of key system resources for both evaluation and troubleshooting purposes. If system performance is suspect but you see no clear indication of what might be slowing it down, the System Monitor can be used to determine which resource is operating at capacity and thereby limiting the performance of the system.

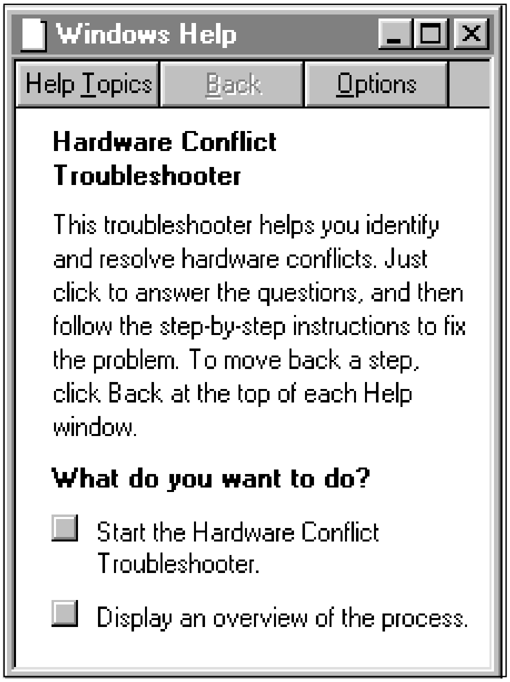


Figure 5.14

Hardware conflict troubleshooter window.

Typical resources that the System Monitor is capable of tracking include those associated with processor usage and memory management. Results can be displayed in real time using statistical mode, line chart mode, or bar chart mode. The System Monitor can be set up to run on top of other applications so that it can be used to see what effect they are having on the system.

The Resource Meter is a simple bar chart display that shows the percent usage of the System Resources, User Resources, and GDI resources. When activated, the meter normally resides as an icon on the extreme right side of the Start Button bar at the bottom of the desktop. Double-clicking on the icon brings the bar chart display to the desktop. As with the System Monitor, the Resource Meter can be used to evaluate hardware and software performance.

Summary

This chapter has covered fundamental troubleshooting tools and techniques. The initial section introduced the basic tools and investigative techniques used to troubleshoot computer systems. This section dealt with the early steps of computer problem solving, including initial steps, differentiating hardware from software problems, and identifying configuration problems.

The second section of the chapter focused on software diagnostic packages and their use. Several actual software diagnostic packages were featured to provide an insight into how they serve different parts of the troubleshooting process.

The diagnostics section was followed by an introductory section for hardware troubleshooting activities. This section included a guide to FRU-style troubleshooting, as well as three specialized hardware troubleshooting procedures.

The final section of the chapter contained special troubleshooting information concerning operating systems. This information includes DOS, Windows 3.x, and Windows 95 startup and operational problems.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand and can perform each item listed there. Afterward, answer the Review questions that follow to verify your knowledge of the information.

Lab Procedures

Hands-on lab procedures correspond to the theory materials presented in this chapter. Refer to the lab manual and perform Procedures 7, "Symptoms," 8, "MSD," 10, "PC-Check," 9, "CheckIt," 18, "Isolating Power Supply Problems," 17, "System Inoperable," 15, "Undefined Problem Isolation," and 11, "Using POST Cards."

Review Questions

1. If the system issues a single beep and the C:\> prompt appears on the screen, what condition is indicated?
2. If an error occurs before the single beep tone in the bootup sequence, what type of failure is probable?
3. If an error occurs after the single beep in the bootup process, what type of problem is likely?
4. If the system refuses to boot up after a new component is installed, what type of problem is normally assumed?
5. List the three important tools used to isolate operating system problems.
6. List three situations that would normally require that the CMOS Setup routines be run.
7. What type of problem is indicated by a "Strike F1 to continue" message during bootup?
8. What is the recommended method of using a digital multimeter to check voltage in a computer system?
9. What component has the capability to affect the operation of all other sections of the computer system?
10. If the system functions correctly after all the optional equipment has been removed, what action should be taken next?
11. If you are replacing components one at a time and the system suddenly begins working properly, what can be assumed?

12. If the system boots up from a Clean Boot disk or an Emergency Startup disk, what action should be taken next?

13. List three items commonly tested using the resistance function of a multimeter.

14. What function and reading would be appropriate for checking a system's speaker?

15. Which file acts as the log file for Windows 3.x problems?

16. What action should be taken first if a software failure is suspected?

17. Which Windows 95 file should be consulted to determine what event led up to a crash during the detection phase of the startup procedure?

18. What resistance reading would normally be expected from a fuse if it is functional?

19. If you are measuring across a capacitor on the system board with a DMM, what voltage reading would you normally expect to see from a DMM?

20. Which noncomputer possibility should be eliminated early in the troubleshooting process?

21. List the files that should be included in a Clean Boot disk.

22. Which Windows 95 file would be consulted to determine at what point a bootup sequence had failed?

23. To what range should the voltage function of a DMM be set for an initial measurement?

24. In the DETLOG sample file, in how many places does the PnP system look for the floppy disk drive controller? Where does it find it and what system resources does it allocate to it?

25. In the detection sample, what function is assigned to the system's IRQ-13 channel?

26. If a "[000E420A] SYSCRITINIT = VMM" line is encountered in the BOOTLOG.TXT file without a following "[000E420A] SYSCRITINITSUCCESS = VMM" line, what can be assumed about the system?

Review Answers

1. The system has successfully booted up to the C:\ DOS prompt. For more information, see the section "Isolating Undefined Problems."

2. If the system produces an error message or beep code signal before the beep, the problem is hardware-related. For more information, see the section "Isolating Undefined Problems."

3. If the error message or beep code is produced after the beep, the problem is likely to be associated with the operating system. For more information, see the section "Isolating Undefined Problems."

4. A setup or configuration problem has occurred. For more information, see the section "Determining Hardware/Software/Configuration Problems."

5. The tools are the `system log files`, `clean boot disks`, and `single-step startup procedures`. For more information, see the section "Operating Systems Troubleshooting."

6. When installing a new system, after replacing the CMOS backup battery, and when a new option is installed in the system. For more information, see the section "Determining Hardware/Software/Configuration Problems."

7. The system has encountered invalid configuration information during the bootup process. Either the configuration has been set incorrectly, or the hardware was unable to confirm the configuration settings. For more information, see the section "Determining Hardware/Software/Configuration Problems."

8. The leads of the meter should be placed so that the meter is in parallel with the device being checked. The tests must be performed while power is applied to the component. For more information, see the section “Using a Multimeter.”

9. The power supply. For more information, see the section “Isolating Power Supply Problems.”

10. Reinstall the removed components one at a time until the problem reappears. The last component returned to the system is at least partially responsible for the problem. However, other removed components must be reinstalled to confirm that only one was bad. For more information, see the section “Hardware Troubleshooting.”

11. That `at least` the last component reinstalled is defective. For more information, see the section “FRU Troubleshooting.”

12. If the system boots up from the minimal condition of the Clean Boot or Startup disk, restart the system and press the F8 function key while the “Starting MS-DOS” message is on the screen to single-step through the CONFIG.SYS and AUTOEXEC.BAT files. The single-step method can be used to isolate the problem command. For more information, see the section “DOS Startup Problems.”

13. Fuses, speakers, and the continuity of connecting cables. For more information, see the section “Using a Multimeter.”

14. Near zero ohms. For more information, see the section “Using a Multimeter.”

15. DRWATSON.LOG. For more information, see the section “Using Dr. Watson.”

16. Test the hardware with software that has been known to operate correctly in the past. For more information, see the section “Determining Hardware/Software/Configuration Problems.”

17. DETLOG.TXT. For more information, see the section “DETCRASH.LOG.”

18. Zero ohms. For more information, see the section “Using a Multimeter.”

19. 5 VDC. For more information, see the section “Using a Multimeter.”

20. The operator. For more information, see the section “Initial Troubleshooting Steps.”

21. The FDISK, FORMAT, SYS, EDIT, CHKDSK, MSD, CONFIG.SYS, and AUTOEXEC.BAT files should be included in the clean boot disk. For more information, see the section “Self-Booting DOS Disk.”

22. The BOOTLOG.TXT file contains the sequence of events conducted during the Startup of the system. For more information, see the section “BOOTLOG.TXT.”

23. The highest range possible. For more information, see the section “Using a Multimeter.”

24. The system checked for the Floppy Controller at IO=3f2-3f5, IO=372-375, and detected it at IO=3f2-3f5. It allocates IRQ=6 and DMA=2 to the floppy drive. For more information, see the section “DETLOG.TXT.”

25. Detected: *PNP0C04\0000 = [6] Numeric data processor; IO=f0-ff; IRQ=13. For more information, see the section “DETLOG.TXT.”

26. The system-critical virtual machine driver has not been successfully initialized. For more information, see the section “BOOTLOG.TXT.”

Chapter 6

System Boards

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Define AT-compatible interrupt and DMA addresses.
- List advantages of chip-set–based circuit design.
- Identify various microprocessors from their package type.
- Discuss the major attributes of popular microprocessors.
- Define real, protected, and virtual addressing modes.
- Define MMX technology.
- Describe the internal structure of a typical integrated peripheral controller (IPC) IC.
- Describe operation of the real-time clock (RTC) circuitry.
- Identify the internal register structure of the IPC's timer/counter subsystem.
- Describe the use of the timer/counters output signals in the typical PC-compatible system.
- List the sequence of events that occur during a typical interrupt operation.
- Identify conditions that will cause an NMI interrupt to occur.
- Describe the events of a typical DMA operation, and differentiate between the modes of DMA transfers.

continues

- Differentiate between static and dynamic RAM memory devices, and state the conditions that dictate which of the two will be used in a given application.
- Differentiate between EVEN and ODD parity schemes.
- Describe how the microprocessor works with a first or second level cache memory.
- Describe the operation of the various subsections that work together to form an advanced architecture system board.
- Describe steps to troubleshoot various system board problems.
- List symptoms and error messages associated with system board problems.

Introduction

The system board contains the components that form the basis of the computer system. Although the system board's physical structure has changed over time, its logical structure has remained relatively constant. Since the original PC, the system board has contained the microprocessor, its support devices, the system's primary memory units, and the expansion slot connectors. Figure 6.1 shows a typical system board layout.

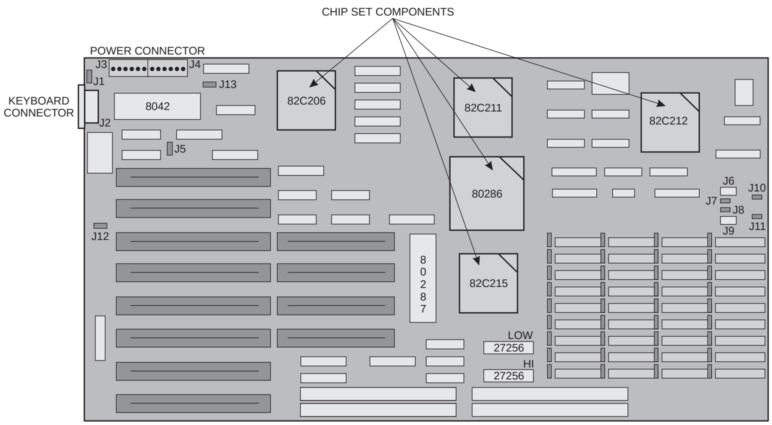


Figure 6.1

A typical system board layout.

System Board Evolution

Objective

System boards fundamentally change for three reasons: new microprocessors, new expansion-slot types, and reduced chip counts. Reduced chip counts are typically the result of improved microprocessor support chip sets. Chip sets combine PC- and AT-compatible structures into larger ICs (integrated circuits). Several IC manufacturers produce single ICs that contain the AT's interrupt, DMA, timer/counter, and real-time clock circuitry. These ICs also contain the address decoding and timing circuitry to support those functions.

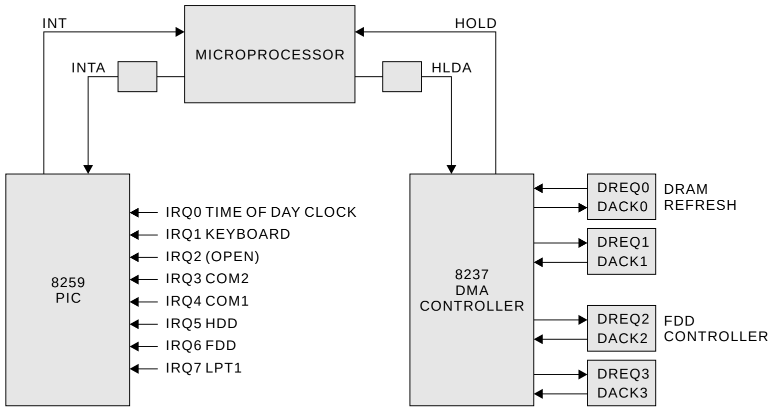
The original IBM PC used a 6-chip chip set to support the 8088 microprocessor. These devices included an 8284 clock generator, an 8288 bus controller, an 8255 parallel peripheral interface (PPI), an 8259 programmable interrupt controller (PIC), an 8237 DMA controller, and an 8253 programmable interval timer (PIT).

An 8042 intelligent keyboard controller completed the PC/XT intelligent support devices. The clock generator and bus controller ICs assisted the microprocessor with system clock and control bus functions. The PPI chip handled system configuration and onboard addressing functions for the system's intelligent devices.

The interrupt controller provided the system with eight channels of programmable interrupt capabilities. The 8237 DMA controller provided four channels of high-speed DMA data-transfer service for the system. The 8253 was used to produce three, programmable timer channel outputs to drive the system's time-of-day clock, DRAM refresh signal, and system speaker output signal. The PC/XT interrupt and DMA controller functions are described in Figure 6.2.

Figure 6.2

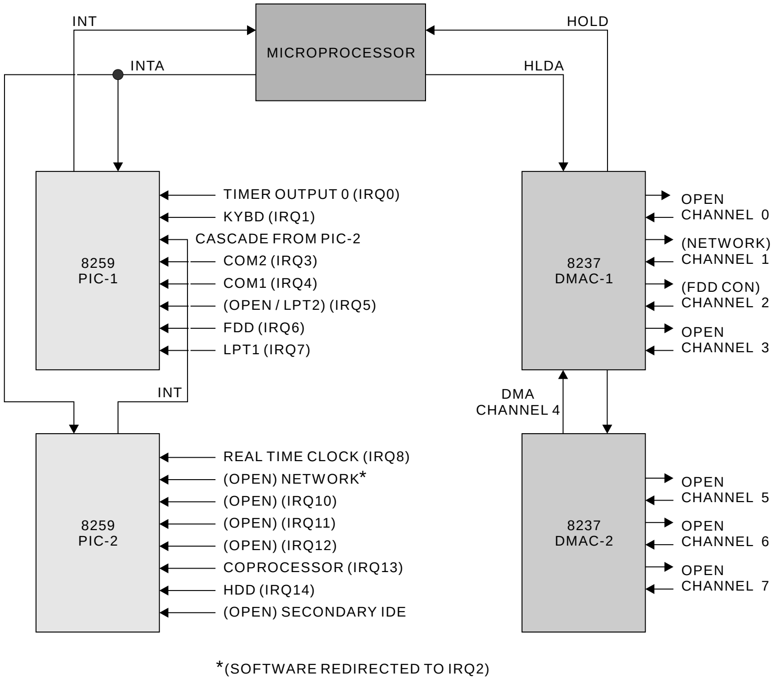
PC/XT Interrupt and DMA controller functions.



When the IBM PC-AT came to the market, it brought an upgraded chip set that expanded the capabilities of the system. IBM improved the basic 8284 and 8288 devices by upgrading them to 82284 and 82288 versions. Likewise, the keyboard controller and the three-channel timer/counter were updated in the AT. The AT's interrupt and DMA channel capabilities were both doubled—to 15 and 7 respectively—by cascading two of each device together. In each case, one IC is the master device and the other is the slave device. The PC-AT interrupt and DMA controller functions are described in Figure 6.3.

Figure 6.3

PC-AT interrupt and DMA controller functions



Chip Sets

Objective

Because chip-set–based system boards require much fewer SSI, MSI, and LSI devices to produce, printed-circuit-board manufacturers have been able to design much smaller boards. The original AT system board measured 30.5×33 cm. Initially, printed-circuit-board manufacturers reduced the size of their chip-set–based system boards to match that of the original PC and PC-XT system boards (22×33 cm). This allowed the new 286 boards to be installed in the smaller XT-style cases. This particular size of system board is described as a baby AT system board.

As VLSI technology improves, IC manufacturers continue to integrate higher levels of circuitry into their chips. All the functions of the 4-chip chip set system board of Figure 6.1 are duplicated using the 2-chip chip set depicted in Figure 6.4. The high level of circuit concentration in this chip set allows the size of the system board to be reduced even further. It is approximately half the length of a standard baby AT system board; therefore, this size system board is referred to as a half-size system board.

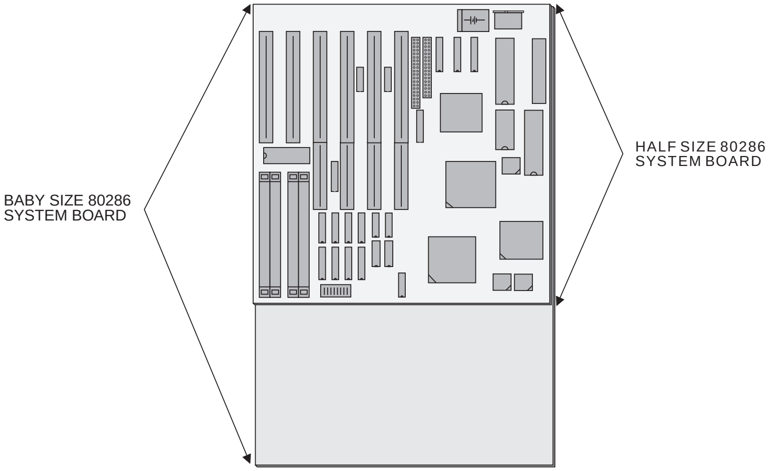


Figure 6.4

A half-size system board.

The first chip-set–based system boards came with special configuration software that had to be loaded from disk. Now, however, it must consider the ROM BIOS as an integral part of the chip set because it is designed to support the register structure of the chip set. Therefore, replacing a ROM BIOS chip on a system board is not as simple as placing another ROM BIOS in the socket. The replacement BIOS must be correct for the chip set being used.

By combining larger blocks of circuitry into fewer ICs, a price reduction spiral is created. Fewer ICs on the board leads to reduced manufacturing costs to produce the board. The material cost of the board is decreased due to its smaller physical size. The component cost is decreased because it is cheaper to buy a few VLSI chips than several SSI or MSI devices. Finally, the assembly cost is less because only a few items must be mounted on the board.

Reduced board costs create lower computer prices, which in turn creates greater consumer demand for the computers. Increased demand for the computers, and therefore the chip sets, acts to further push down the prices of all the computer components.

Chip-set–based system boards and I/O cards tend to change often as IC manufacturers continue to integrate higher levels of circuitry into their devices. The newest system board designation is the ATX form factor developed by Intel for Pentium-based systems. This specification is an evolution of the baby AT form factor that moves the standard I/O functions to the system board. The ATX specification basically rotates the baby AT form factor by 90 degrees, relocates the power supply connection, and moves the microprocessor and memory modules away from the expansion slots.

The power supply orientation allows a single fan to be used to cool the system. This provides reduced cost, reduced system noise, and improved reliability. The relocated microprocessor and memory modules allow full-length cards to be used in the expansion slots while providing easy upgrading of the microprocessor, RAM, and I/O cards.

Figure 6.5 depicts a Pentium-based, ATX system board that directly supports the FDD, HDD, serial, and parallel ports. The board is 12 inches (305 mm) wide and 9.6 inches (244 mm) long. A revised, mini-ATX specification allows for 11.2×8.2-inch system boards. The hole patterns for the ATX and mini-ATX boards require a case that can accommodate the new boards. Although ATX shares most of its mounting hole pattern with the baby AT specification, it does not match exactly.

The fully implemented ATX format also contains specifications for the power supply and I/O connector placements. In particular, the ATX specification for the power supply connection calls for a single 20-pin power cable between the system board and the power supply unit rather than the typical P8/P9 cabling. The new cable adds a +3.3 Vdc supply to the traditional +/- 12 Vdc and +/- 5 Vdc supplies. In most monitors, the frequency relationships between these signals are fixed. However, there are some monitors, referred to as `Multisync monitors`, that can adapt to various horizontal sweep rates and vertical refresh rates to accommodate a variety of different video standards. Some of these standards are discussed later in this chapter (see Figure 9.20).

System boards designed for use in notebooks typically include the video circuitry as an integral part of the board. These units must also provide the physical connections for the unit's parallel and serial I/O ports, as well as onboard connectors for auxiliary disk drive, monitor, and keyboard units.

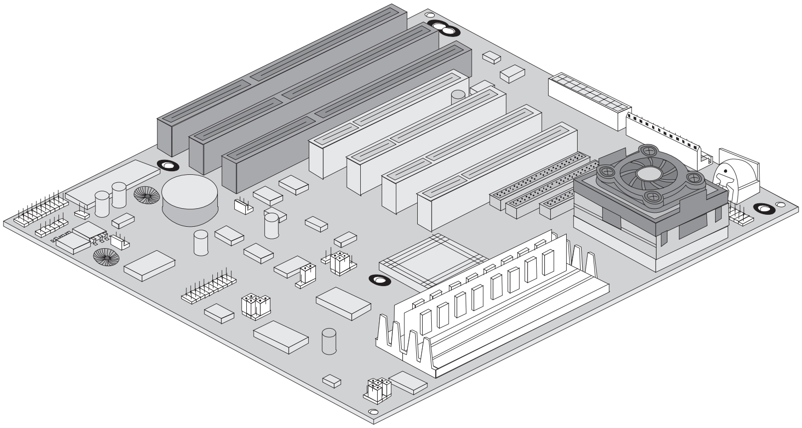


Figure 6.5

An ATX Pentium system board.

Microprocessors

Objective

The microprocessors used in the vast majority of all PC-compatible microcomputers include the 8088/86, the 80286, 80386, 80486, and Pentium (80586 and 80686) microprocessors. The 8088 material that follows provides the background for the microprocessors that have been used to advance the PC-compatible line of microcomputers to the computational levels they now possess. The popularity of the original PCs (and the software developed for them) has caused limitations to be built in to the microprocessors that followed the 8088 to maintain compatibility with it.

The 8088 Microprocessor

At the heart of PC- and PC-XT–compatible systems is the 40-pin 8088 microprocessor. The 8088 is a high-performance HMOS microprocessor which has attributes of both 8- and 16-bit processors. Its internal structure supports 16-bit words, but it uses only an 8-bit data bus. In this manner, its internal data registers can be used as single 16-bit registers, divided into higher- and lower-order bytes, or used as independent 8-bit register pairs.

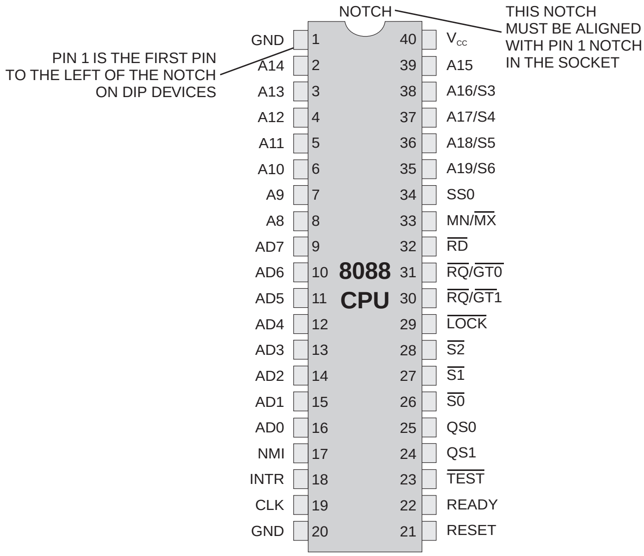
The 8088 supports a 20-bit address bus that allows it to directly access 1MB of memory and I/O addresses. The 8088-2 version can run at either of two standard clock rates: a 4.77-MHz normal clock frequency and a high-speed (Turbo) 8-MHz clock frequency.

8088 Pin Assignments

The pin assignments and definitions of the 8088 microprocessor are described in Figure 6.6. To hold the number of pins on the IC down to a reasonable level, the 8088's designers `multiplexed` many of its pins to perform different functions, depending on which part of the machine cycle the microprocessor is in.

Figure 6.6

8088 pin assignments.



The most notable of the multiplexed pins are the address/data lines (AD0–AD7) and the address/status pins (A16/S3–A19/S6). Pins AD0 through AD7 are time-multiplexed to act as address lines during the first clock pulse of the machine cycle, and then turn into bidirectional data lines throughout the remainder of the machine cycle. Likewise, the address/status pins are multiplexed to act as address lines during the first clock pulse and then change to processor status lines for the remainder of the machine cycle.

Collectively, the AD0–AD7, A8–A15, and A16/S3–A19/S6 lines are combined to form the system's address bus. The 8088 uses the entire 20 bits when it is addressing memory locations, but only lines A0 through A15 are used when addressing input/output devices. This provides a total of 1,048,576 bytes (1MB) of memory addresses and 65,535 bytes (64KB) of I/O addresses.

After the address has been latched, the AD0–AD7 lines are available for bidirectional data transfers. The computer system uses a bidirectional line buffer to assist the 8088 in driving the system's data bus. A pair of signals from the bus controller controls the transfer of data through the line buffer. The bus controller issues a `data enable signal` (DEN) to activate the buffer, along with a `data transmit/receive direction signal` (DT/R), which properly configures the buffer so that the microprocessor can read (receive) or write (transmit) through them. Figure 6.7 illustrates how the 8088's address/data/status pins are multiplexed to form the address and data buses.

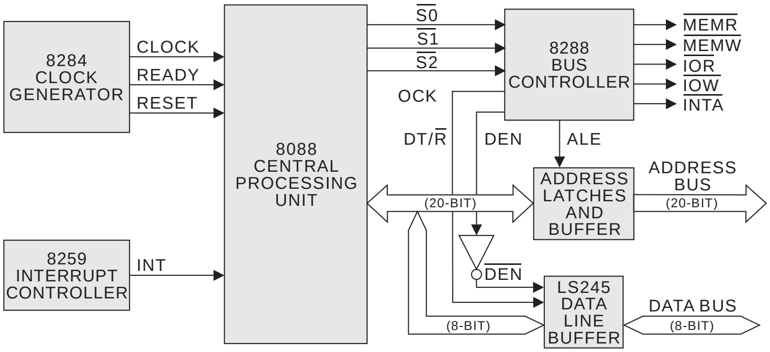


Figure 6.7

8088 bus multiplexing.

8088 Register Set

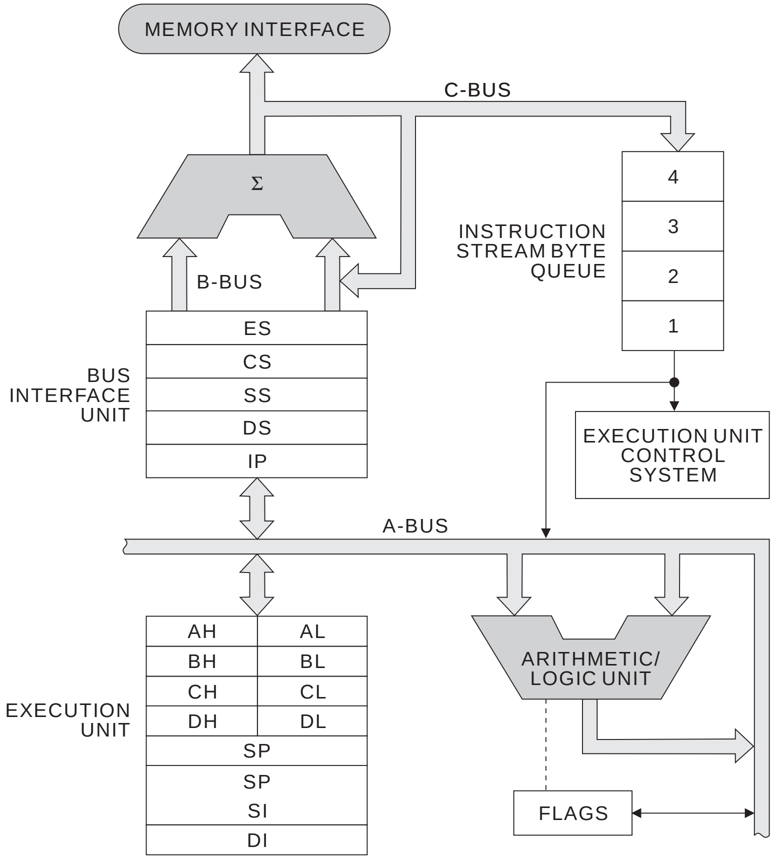
Figure 6.8 illustrates the basic architecture of the 8088 in block diagram form. One of the most noticeable features of the 8088's internal structure is its large and very flexible register set. The register set consists of 14 16-bit registers which can be divided into four functional groups, as follows: the data group, the index and pointer registers, the segment registers, and the instruction pointer and flag registers.

The most interesting of these registers is the data group (AX, BX, CX, and DX). These highly flexible registers are unique in that each can be treated as a single 16-bit register, or they may be used as two, independent 8-bit registers (AH and AL, BH and BL, CH and CL, and DH and DL). The "H" and "L" designators refer to the higher- and lower-order bytes of the basic 16-bit register. All these registers correspond roughly to the accumulator register described in Chapter 1. Arithmetic and logic operations can be performed, and results can be stored in each of the data registers.

(These operations can also be performed in the two index and pointer registers.)

Figure 6.8

Inside the 8088.



8088 Addressing

You may wonder how the 8088 can support a 20-bit address bus if its internal registers are only 16 bits wide. Actually, the 20-bit address is constructed inside the 8088 by combining the contents of two registers. The 8088's memory addresses are divided into 64-KB blocks called `segments`. These segments can be assigned to the segment registers in the 8088. Within each 64-KB segment, individual addresses can be accessed using only a 16-bit address called the `offset address`. To obtain the entire 20-bit address, the 16-bit offset address is added to a 16-bit segment address, which has been shifted left four binary places. This concept is illustrated in Figure 6.9.

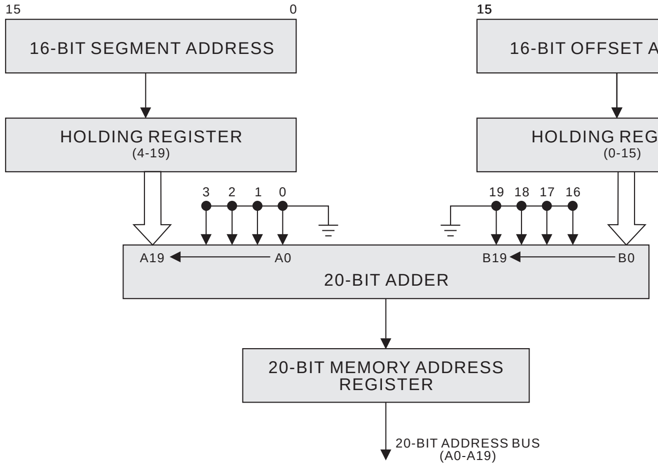


Figure 6.9

8088 addressing.

During instruction cycles, the 8088's address is the sum of the instruction pointer (IP) and code segment (CS) registers. During execution cycles, the address produced is the sum of the operand address portion of the instruction word and the contents of one of the segment registers (DS, SS, or ES).

In special addressing operations involving the stack, the contents of the stack segment (SS) register is added to the contents of either the base pointer (BP) or the stack pointer (SP) registers to obtain the address. The data segment (DS) register is used whenever data is moved to or from the memory.

Normally, these registers and their use are more important to a programmer than to a technician. The reason for mentioning the 8088's addressing here is to illustrate how the address is generated. In many publications and diagnostic software packages, addressing is presented in an XXXX:YYYY format (that is, FFFF:0000). The reason for this is found in the segmented nature of how the 8088 viewed addressing. The first set of numbers represent the segment value, and the number to the right of the colon is the offset value.

The 80286 Microprocessor

Objective

One of the key elements of the IBM-PC AT system board is the 80286 16-bit microprocessor chip. The 80286 is backward compatible with the 8088 microprocessor found on the original PC and XT system boards. This means that the 80286 will run the same software that the 8088 does, but it will run it much faster. However, the 80286 is much more than a fast 8088 microprocessor.

Unlike the 8088 (or its full 16-bit equivalent—the 8086), the 80286 microprocessor is designed to support `multi-user` and `multi-tasking` operations. In these types of operations, the 80286-based machine appears to work on several tasks, or to serve several users simultaneously. Of course, the microprocessor can't actually work on more than one item at a time; the appearance of simultaneous operations is created by storing the parameters of one task, leaving the task, loading up the state of another task, and beginning operation on it.

Figure 6.10 shows the 80286 microprocessor in block-diagram form. The special memory management and memory protection circuitry in the address unit (AU) of the 286 handles task separation, program and data integrity between tasks, and isolation of the operating system from the task-switching operation. When two or more programs are running together in protected mode, this circuitry keeps track of the different tasks that the 286 is working on, and keeps them from interfering with one another's data or program space.

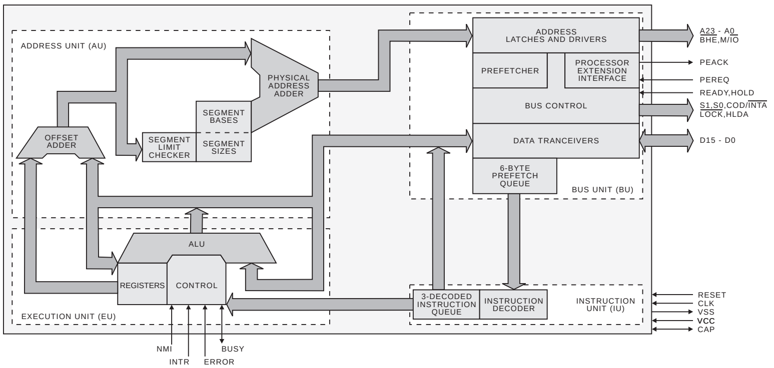


Figure 6.10

Inside the 80286.

As its name implies, the `Bus Interface Unit` (BU) controls the action between the 286's internal structures and the system's local buses. In addition, the BU sets priorities between the `Code Prefetch Unit` (CU) and the `Execution Unit` (EU). Execution Unit operations (where instructions are actually processed) are given the highest priority by the BU. However, the BU allows the code Prefetch unit to sequentially fetch instructions from memory during spare clock cycles. The Prefetch Unit stores the instructions in a 6-byte queue, awaiting decoding and execution.

The `Instruction Decoding Unit` (IU) decodes instruction words and stores them in a three-instruction, first-in first-out (FIFO) queue. In this way, the Execution Unit does not have to wait for a new instruction to be fetched from memory, because the instruction is already waiting in the decoded instruction queue.

The 80286's internal register set is shown in Figure 6.11. As you can see, it is identical to the register set found in the 8088. However, the 286 possesses an extended 8088/86 instruction superset that allows it to operate in two distinctly different addressing modes; these are real mode and virtual-protected mode.

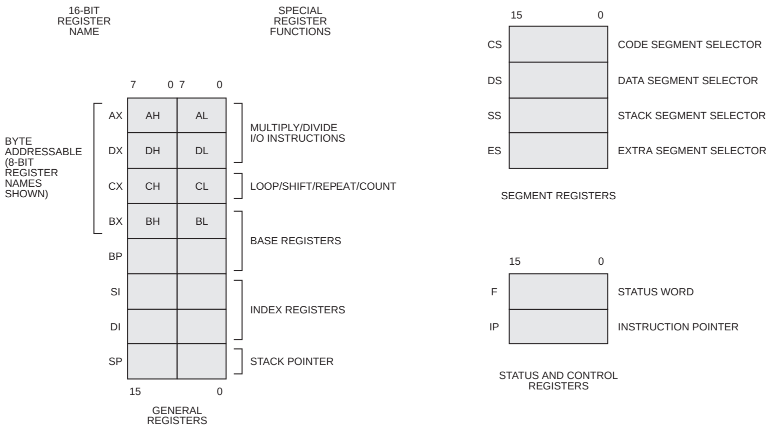


Figure 6.11

The 80286 register set.

In real mode, the 286 operates like an 8088/86 microprocessor, and can directly access 1MB of RAM addresses in segments of 64 KB. It also uses real addresses up to 1MB, and can work on only one task at a time. In this mode, addresses are specified by the bits present on the 80286's A0–A19 pins. If software increments the 80286's addresses past FFFFFh in this mode, the address will just roll over to 00000h and address bits A20–A23 will not be activated.

In protected mode, address bits A20–A23 are enabled, and the 286 can access up to 16MB of physical memory addresses. It can also be used to perform virtual memory operations. In these operations, the system treats an area of disk space as an extension of RAM memory. It can shift data from RAM memory to disk (and vice versa) to simulate large areas of RAM (up to 1024MB, or 1GB). The 286 requires external support circuitry and special software, however, to conduct virtual memory operations. Using protected mode addressing, the 80286 maps up to 1GB (2^{30}) of memory into an actual address space of 16MB (2^{24}).

The 80286 is available in 68-pin Ceramic Leadless Chip Carrier (CLCC), 68-pin Plastic Leaded Chip Carrier (PLCC), and 68-pin Pin Grid Array (PGA) packages, as depicted in Figure 6.12. These microprocessors are also available with different clock/speed ratings. A microprocessor marked 80286-12, for example, is rated for stable operation with internal clock signals up to 12MHz. (The 286 divides the external clock input by a factor of two to arrive at this value. Therefore, the maximum external clock frequency for the 286-12 is 24MHz.)

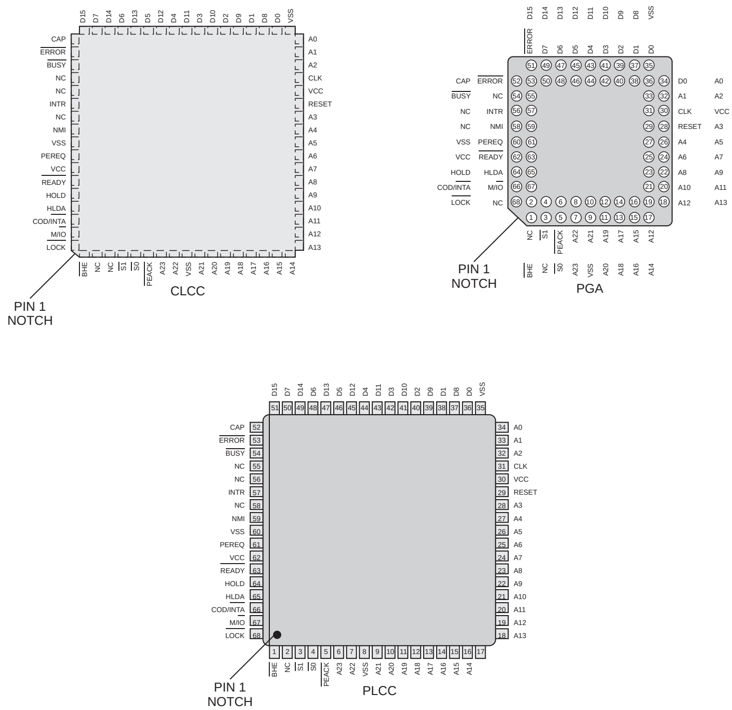
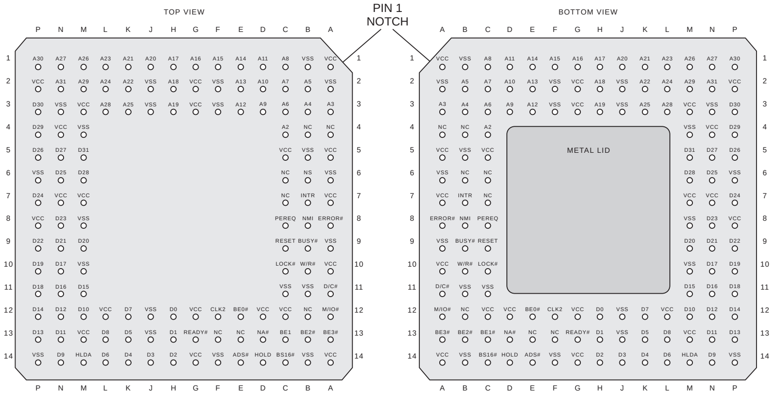


Figure 6.12

The 80286 microprocessor packages.

The 80386 Microprocessor

The 80386DX (or just 80386) microprocessor is the 32-bit successor of the 80286. This microprocessor improves on previous 80x86 architectures by offering 32-bit registers as well as 32-bit address and data buses. The 80386 is produced in the 132-pin, PGA package shown in Figure 6.13.



SIGNAL/PIN		SIGNAL/PIN		SIGNAL/PIN		SIGNAL/PIN		SIGNAL/PIN		SIGNAL/PIN	
A2	C4	A24	L2	D6	L14	D28	M6	VCC	C12	VSS	F2
A3	A3	A25	K3	D7	K12	D29	P4		D12		F3
A4	B3	A26	M1	D8	L13	D30	P3		G2		F14
A5	B2	A27	N1	D9	N14	D31	M5		G3		J2
A6	C3	A28	L3	D10	M12	D/C#	A11		G12		J3
A7	C2	A29	M2	D11	N13	ERROR#	A8		G14		J12
A8	C1	A30	P1	D12	N12	HLDA	M14		L12		J13
A9	D3	A31	N2	D13	P13	HOLD	D14		M3		M4
A10	D2	ADS#	E14	D14	P12	INTR	B7		M7		M8
A11	D1	BE0#	E12	D15	M11	LOCK#	C10		M13		M10
A12	E3	BE1#	C13	D16	N11	M/IO#	A12		N4		N3
A13	E2	BE2#	B13	D17	N10	NA#	D13		N7		P6
A14	E1	BE3#	A13	D18	P11	NMI	B8		P2		P14
A15	F1	BS16#	C14	D19	P10	PEREQ	C8		P8	W/R#	B10
A16	G1	BUSY#	B9	D20	M9	READY#	G13	VSS	A2	N.C.	A4
A17	H1	CLK2	F12	D21	N9	RESET	C9		A6		B4
A18	H2	D0	H12	D22	P9	VCC	A1		A9		B6
A19	H3	D1	H13	D23	N8		A5		B1		B12
A20	J1	D2	H14	D24	P7		A7		B5		C6
A21	K1	D3	J14	D25	N6		A10		B11		C7
A22	K2	D4	K14	D26	P5		A14		B14		E13
A23	L1	D5	K13	D27	N5		C5		C11		F13

Figure 6.13

The 80386DX microprocessor.

Ironically, most 80386 microprocessors were used in systems with AT-compatible architectures. Therefore, its data bus had to support a 16-bit I/O channel. However, operations that were confined to the system board, such as memory reads and writes, could be carried out as full 32-bit transfers. The 80386 microprocessor can actually conduct data transfers in three distinct ways; it can transfer data in 8-bit `Bytes`, in 16-bit `Words`, or in 32-bit `Dwords` (Double words).

In addition to handling data in one of these three basic definitions, the 80386 possesses two methods of manipulating memory units in very large blocks. These memory units are called `pages` and `segments`. The 80386 has the capability to divide memory into one or more variable-length segments that can be swapped to and from disk, or shared between different programs.

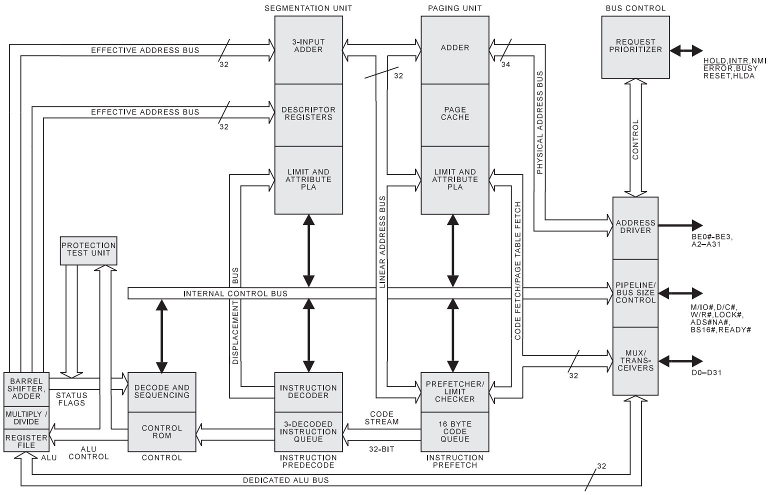
The 80386 has built-in circuitry that can be used to organize RAM memory into one or more 4-KB sections—referred to as `pages`. This memory management scheme effectively divides programs into multiple, uniform-sized modules that can be manipulated as individual blocks, and is particularly useful in virtual memory multitasking operations.

The advanced memory mapping capabilities of the 80386 allow it to implement `ROM shadowing`. This speeds up system operation by allowing BIOS-related functions to be conducted from fast 32-bit RAM locations rather than slower 8-bit ROM locations.

Figure 6.14 depicts the 80386's internal structure in block-diagram form. It illustrates the 80386's six functional units: the Bus Interface Unit (BU), Code Prefetch Unit (CU), the Instruction Decoding Unit (IU), the Execution Unit (EU), the Segmentation Unit (SU), and the Paging Unit (PU). By comparing these blocks to those of the 80286 microprocessor, it can be seen that several of them are carry-overs from the 286.

Figure 6.14

Inside the 80386DX.

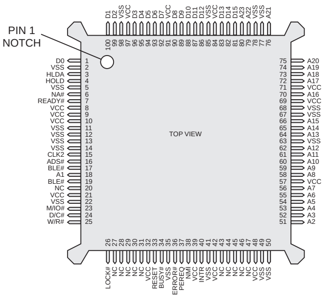


In addition to offering the 80286 version of protected-mode addressing, the 80386 offers an improved protected addressing mode, referred to as `virtual 86 mode`. This mode allows the 80386 to simulate several 8086 microprocessors running at the same time. Each 8086 environment is referred to as a `virtual 86 machine`. Subsequently, the 80386 can run several applications simultaneously. It uses paging to remap memory and bring each virtual machine to the microprocessor as the program running in it requires attention.

The 80386DX possesses an internal divide-by-2 clock system. Therefore, the maximum clock frequency applied to the chip is twice its rated operational frequency (i.e., the maximum external clock frequency of an 80386DX-25 is 50MHz.)

The 80386SX Microprocessor

The 80386SX microprocessor, depicted in Figure 6.15, is a 16/32-bit hybrid version of the 80386DX. It was developed to achieve a compromise between the power of the 80386DX and the lower cost of the 80286 microprocessors. The 80386SX is constructed by using high-speed CHMOS III technology and packaged in a surface-mount, 100-pin, plastic, quad flat pack package.



ADDRESS		DATA		CONTROL		N/C	VCC	VSS
A1	18	D0	1	ADS#	16	20	8	2
A2	51	D1	100	BHE#	19	27	9	5
A3	52	D2	99	BLE#	17	28	10	11
A4	53	D3	96	BUSY#	34	29	21	12
A5	54	D4	95	CLK2	15	30	32	13
A6	55	D5	94	D/C#	24	31	39	14
A7	56	D6	93	ERROR#	36	43	42	22
A8	58	D7	92	HLDA	3	44	48	35
A9	59	D8	90	HOLD	4	45	57	41
A10	60	D9	89	INTR	40	46	69	49
A11	61	D10	88	LOCK#	26	47	71	50
A12	62	D11	87	M/IO	23		84	63
A13	64	D12	86	NA#	6		91	67
A14	65	D13	83	NMI	38		97	68
A15	66	D14	82	PEREQ	37			77
A16	70	D15	81	READY#	7			78
A17	72			RESET	33			85
A18	73			W/R#	25			98
A19	74							
A20	75							
A21	76							
A22	79							
A23	80							

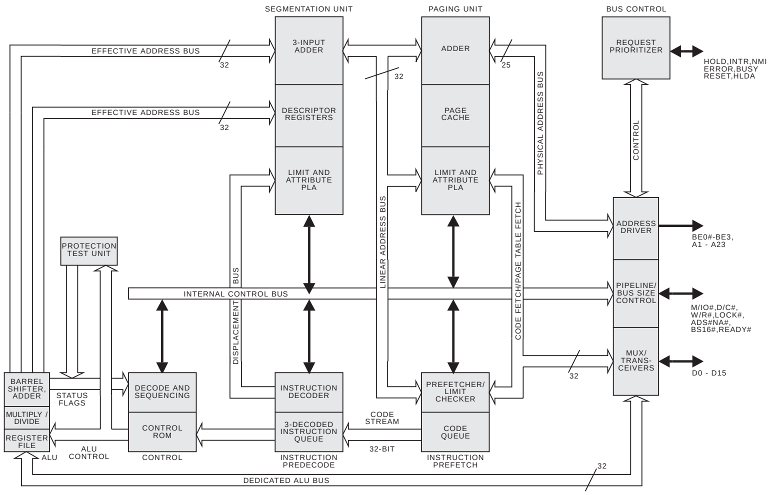
Figure 6.15

The 80386SX microprocessor.

On the inside, the 80386SX is, for all practical purposes, identical to the 80386DX chip. Compare the 80386SX's block diagram, shown in Figure 6.14, with that of the 80386DX in Figure 6.16. As you can see, they appear identical. The major difference between these microprocessors is found in their external address and data buses. Although the 80386SX's internal registers are 32 bits wide (like the 80386DX), it does not support a 32-bit, external data bus. Instead, it supports a 16-bit external data bus.

Figure 6.16

Inside the 80386SX microprocessor.



This makes the 80386SX readily adaptable to the 16-bit architectures originally designed for 80286 microprocessors. Because of the register/bus-size mismatch created by this arrangement, the 80386SX must handle data in a High-byte/Low-byte register format for reading or writing data. This is similar to the 16/8-bit, register/data bus format found in the 8088 microprocessor.

Likewise, the 80386SX's address bus is also smaller than that of the 80386DX. The 80386SX supports a 24-bit address bus. Both processors use the same logical address space, but the 80386SX is limited to only 16MB of physical address space, as opposed to the 4GB available to the 80386DX.

By incorporating these few differences into its makeup, the 80386SX can offer all the performance benefits associated with 32-bit programming, while providing the cost savings associated with 16-bit computer systems. The 80386SX is 100% software compatible with the 80386DX, 286, and 8088/86 microprocessors. Therefore, systems based on it can access and run software already developed for these popular systems.

Other 80386 Microprocessors

Because of the 80386 microprocessor's popularity, other manufacturers have produced similar microprocessors based on the 80386 architecture without infringing on the Intel design. The following are some of the other 80386 variations that you may encounter.

The 80386SL microprocessor is used in laptop or notebook computers. This is due to the lower power consumption of this chip compared to Intel's other 80386s. It is a mixture of the 80386DX and 80386SX microprocessors. This processor retains the 32-bit address bus, like the 80386DX, but has an external 16-bit data bus. The 80386SL still has an internal 32-bit data path, like either the 80386DX or the 80386SX. The 80386SL also keeps the virtual memory capability of the 80386DX or 80386SX.

The AM80386DX was the first `clone` microprocessor. It is manufactured by Advanced Micro Devices (AMD) and was released in early 1991. The AM80386DX mimics the operation of Intel's 80386DX exactly. This includes the 32-bit address and data buses and addressable virtual memory. The AM80386SX is Advanced Micro Devices' release to compete with Intel's 80386SX. This processor is actually more like Intel's 80386SL microprocessor in architecture. This chip also has the 32-bit address bus and a 16-bit external data bus. The AM80386SX does have one advantage with a lower power consumption than Intel's 80386SX.

The IBM 80386SLC microprocessor is the result of a joint venture between Intel and IBM, for exclusive use in IBM products. This chip is actually similar to Intel's 80386SX, which has a 24-bit address bus and a 16-bit external data bus. The enhancements implemented in this design include an 8KB cache and an increased instruction set.

The 80486 Microprocessor

The 80486 (486) microprocessor is the successor to the 80386DX. The block diagram of the 80486 is presented in Figure 6.17. Basically, the 80486 brings an improved 80386DX microprocessor, a high-performance 80387 coprocessor, and an 82385 cache memory controller together in a single package.

Both of these additions greatly improve the speed at which the microprocessor can manipulate numbers. The 80486 also incorporates an onboard 8-KB memory cache, as well as internal parity-generation and checking circuitry.

All this circuitry is combined into a single, 168-pin, Ceramic Pin Grid Array package. Even though most of the overall package is used for heat dissipation purposes, the 80486 still generates so much heat that additional fans are often required. A number of companies have developed snap-on and stick-on fan units for 80486 processors. These units typically derive power from one of the system power supply's auxiliary power connectors.

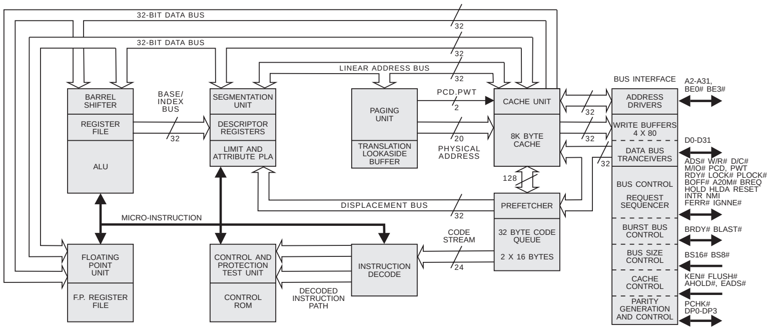


Figure 6.17

Inside the 80486 microprocessor.

Naturally, the 80486 offers the three addressing modes (real, protected, and virtual 86) that the 80386 does. It can access 4GB (2^{32}) of physical addresses and 64TB (2^{46}) of virtual address space. The 80486's paging mechanism allows the 20-bit linear address produced by the virtual mode program to be divided into up to 256 pages. Each page can be located anywhere within the 4GB of the 80486 physical addressing space. An additional I/O protection feature allows the operating system to set aside a selected set of I/O ports for device protection.

The 80486 microprocessor offers vastly improved memory access and instruction execution speed over the 80386DX. It also greatly increases floating-point calculation speed due to the presence of the onboard coprocessor unit. Transfers between the 80486's ALU and coprocessor unit are carried out in the form of 64-bit words.

When the 80486 performs memory read operations that require more than a single bus cycle to complete, it can shift into a special high-speed `burst mode`. Recall from the discussion of 80386 bus cycles that a normal bus cycle is made up of at least two bus states. In burst mode, the normal two-state bus cycle is reduced to one clock cycle after the initial two-state read cycle.

Burst mode is most useful when the 80486 is filling its internal cache memory from the system memory. The cache is updated each time the microprocessor performs a memory read (I/O read operations are not recorded in the cache). When the 80486 performs a memory read operation, it first checks the cache to see whether the requested data is there. If the data is in the cache, no memory is read (and, therefore, no external bus access) is required. This makes for a very fast read operation. If the data is not in the cache, however, the 80486 reads the actual memory location, and then places the data in the cache.

Conversely, all 80486 memory-write operations are performed at the system memory, even if the data is already in the cache. If the bus is busy when the 80486 needs to write data to memory, it can store the data in one of its four onboard write registers. This allows the microprocessor to continue processing internally and wait for a convenient time to write the data into memory. IO-write operations may also be stored temporarily inside the 80486, but, unlike memory-write buffering, multiple IO-writes must be written out to memory before other internal processing can continue.

The 80486's internal cache controller monitors the system's address bus when other processors or bus masters gain control of the bus system. The reason for this is to keep track of addresses where new data may be written into memory, but not into the cache. By keeping track of these address locations, the cache's controller can update the cache as soon as the 80486 regains control of the buses. Many manufacturers of 80486-based systems opt to include additional, external cache memories on their system boards.

In these systems, the 80486's internal cache memory is referred to as the `first-level` cache, and the external cache is called the second-`level` cache. Secondary caches are normally 128KB or 256KB.

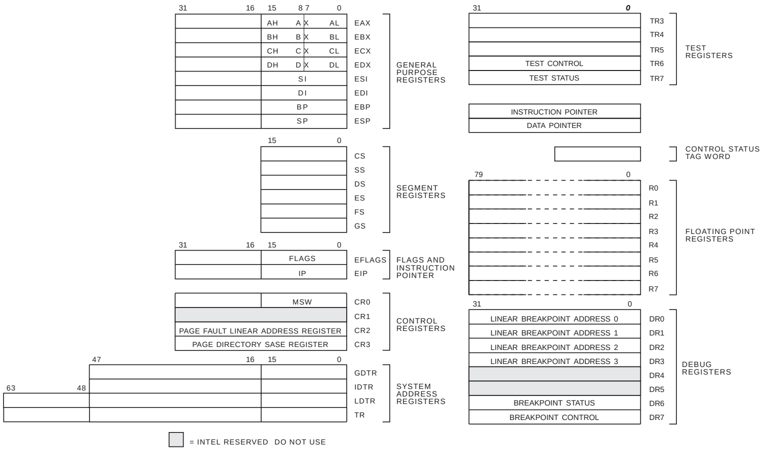
When the 80486 writes data into memory, it generates a parity bit for each byte. As in other systems, the parity bit is stored in system

RAM along with the data. When the data is read from memory, the parity bit is also read. If an error condition is detected by the 80486, it activates its PCHK output line to notify the system. It is the responsibility of the system's decision circuitry to handle the error condition. The 80486's parity check circuitry checks only information retrieved from RAM, not internally generated data.

Excluding its floating-point coprocessor's registers, the 80486's general register structure, described in Figure 6.18, is very similar to that of the 80286 and 80386 processors. These registers can be divided into three distinct groups: the base architecture registers, the system level registers, and the floating-point registers. The 80486's basic register set is identical with those in the 80386 processor. Registers are 32 bits wide, but can be accessed in 8- or 16-bit formats.

Figure 6.18

The 80486 register structure.

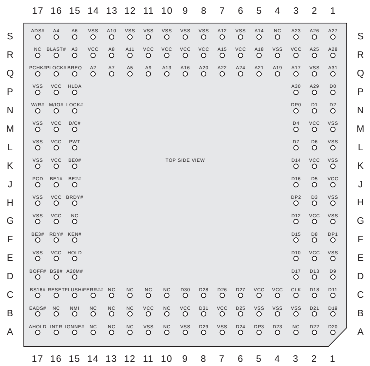


Likewise, the floating-point registers are identical to those of the 80387 coprocessor. The instruction and data pointer registers perform the same functions as their 387 counterparts in determining the cause of coprocessor exceptions.

The 80486's pin groupings are displayed in Figure 6.19. Most of these lines should be familiar to you from previous microprocessor discussions. Naturally, the coprocessor interface lines used with previous microprocessors have been discarded on the 80486. However, new lines have been added for cache and burst-mode control, as well as to provide parity generation and checking functions.

Figure 6.19

The 80486 pins.



ADDRESS		DATA		CONTROL		N/C	VCC	VSS
A2	Q14	D0	P1	A20M#	D15	A3	B7	A7
A3	R15	D1	N2	ADS#	S17	A10	B9	A9
A4	S16	D2	N1	AHOLD	A17	A12	B11	A11
A5	Q12	D3	H2	BE0#	K15	A13	C4	B3
A6	S15	D4	M3	BE1#	J16	A14	C5	B4
A7	Q13	D5	J2	BE2#	J15	B10	E2	B5
A8	R13	D6	L2	BE3#	F17	B12	E16	E1
A9	Q11	D7	L3	BLAST#	R16	B13	G2	E17
A10	S13	D8	F2	BOFF#	D17	B14	G16	G1
A11	R12	D9	D1	BRDY#	H15	B16	H16	G17
A12	S7	D10	E3	BREQ#	Q15	C10	J1	H1
A13	Q10	D11	C1	BS8#	D16	C11	K2	H17
A14	S5	D12	G3	BS16#	C17	C12	K16	K1
A15	R7	D13	D2	CLK	C3	C13	L16	K17
A16	Q9	D14	K3	D/C#	M15	G15	M2	L1
A17	Q3	D15	F3	DP0	N3	R17	M16	L17
A18	R5	D16	J3	DP1	F1	S4	P16	M1
A19	Q4	D17	D3	DP2	H3		R3	M17
A20	Q8	D18	C2	DP3	A5		R6	P17
A21	Q5	D19	B1	EADS#	B17		R8	Q2
A22	Q7	D20	A1	FERR#	C14		R9	R4
A23	S3	D21	B2	FLUSH#	C15		R10	S6
A24	Q6	D22	A2	HLDA	P15		R11	S8
A25	R2	D23	A4	HOLD	E15		R14	S9
A26	S2	D24	A6	IGNNE#	A15			S10
A27	S1	D25	B6	INTR	A16			S11
A28	R1	D26	C7	KEN#	F15			S12
A29	P2	D27	C6	LOCK#	N15			S14
A30	P3	D28	C8	M/IO#	N16			
A31	Q1	D29	A8	NMI	B15			
		D30	C9	PCD	J17			
		D31	B8	PCHK#	Q17			
				PWT	L15			
				PLOCK#	Q16			
				RDY#	F16			
				RESET	C16			
				W/R#	N17			

The rated clock input to the 80486 is different from that of the previous 80X86 microprocessors in that there is no internal division factor to arrive at the chip's speed rating. In other words, an 80486-25 microprocessor must operate from a 25-MHz clock input signal. Conversely, a 16-MHz 80386SX requires a clock signal of 32 MHz to operate properly. In addition, there is no provision for slowing the 80486 microprocessor down during operation to accommodate expansion bus usage (normal/turbo operations). This may adversely affect software that relies on clock-dependent timing loops to accomplish tasks.

Other 80486 Microprocessors

The 80486SX microprocessor, produced by Intel, is an 80486DX version that does not have the math coprocessor built in. This relates to a lower cost for the computer system. Most of the systems that use the 80486SX have the capability of adding a separate math coprocessor if desired. The math coprocessor for the 80486SX is identified as an 80487SX. The 80486SX maintains the same bus sizes as the 80486DX.

The 80486DX2 is identical to the 80486DX in operation and bus size. However, the 80486DX2 adds a feature called `clock doubling`. This technology, introduced by Intel, gives the appearance of doubling the frequency of the clock internally to decrease execution time. The external bus operation, however, is still at the speed determined by the input-clock frequency. This feature is actually achieved by increasing the internal efficiency of the chip design. Therefore, an 80486DX2 with an input-clock frequency of 33MHz has the appearance of the same chip operated at 66MHz.

The 80486DX4 microprocessor, also produced by Intel, takes the clock-doubling technology to the next level to produce a `clock-tripling` design. This, again, is achieved by increasing the internal efficiency of the architecture.

The Cyrix CX486SLC is technically more like an 80386SX microprocessor in architecture. It has a 24-bit address bus and a 16-bit external data bus. The math coprocessor is also not included in this design. The Cyrix CX486DLC microprocessor is Cyrix's version of Intel's 80486DX. Like the 80486DX, this chip includes the 32-bit address and data buses and math coprocessor. The CX486DRU2 is Cyrix's answer to Intel's 80486DX2. This chip offers the same architecture and features as the 80486DX2.

The design of the 80486SLC2 is IBM's version of an 80486DX2 and an 80386SX. This chip has the bus size of the 80386SX and the "clock-doubling" technology of the 80486DX2 integrated in one package. This design does not include a math coprocessor. However, it does use the 80486 instruction set.

The Pentium Processor

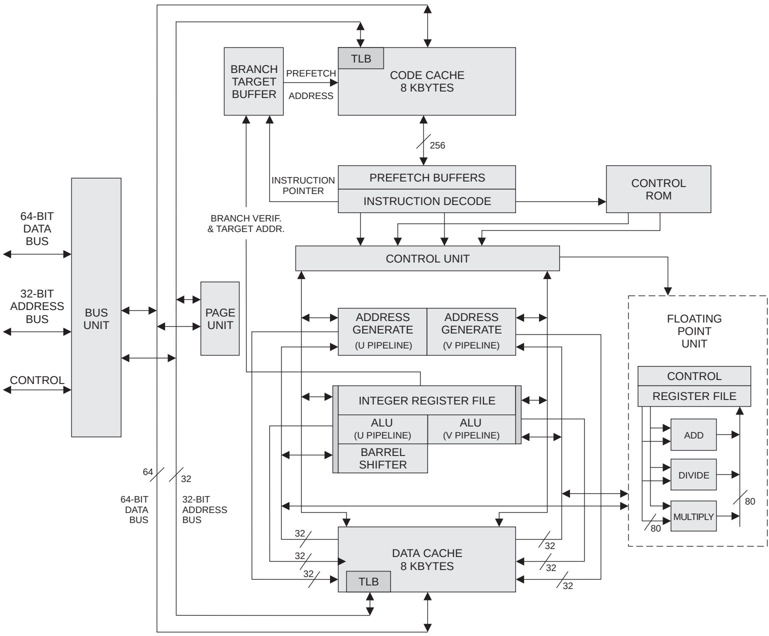
The Pentium processor succeeds the 80486 microprocessor, and maintains compatibility with other 80X86 microprocessors. The Pentium is a 32/64-bit microprocessor contained in a 273-pin PGA package. Like the 80486 package, the Pentium generates so much heat during normal operation that an additional CPU cooling fan is required.

All registers for the microprocessor and floating-point sections of the Pentium are identical to those of the 80486. The Pentium has a 64-bit data bus that allows it to handle `Quad word` (or Qword) data transfers. The Pentium also contains two separate 8-KB caches, compared to only one in the 80486. One of the caches is used for instructions or code, and the other is used for data. The internal architecture of the Pentium resembles an 80486 in expanded form. The floating-point section operates up to five times faster than that of the 80486.

The internal architecture is shown in Figure 6.20. The Pentium is called a `superscalar` microprocessor because its architecture allows multiple instructions to be executed simultaneously. This is achieved by a `pipelining` process. Pipelining is a technique that uses multiple stages to speed instruction execution. Each stage in the pipeline performs a part of the overall instruction execution, with all operations being completed at one stage before moving on to another stage. This technique allows streamlined circuitry to perform a specific function at each stage of the pipeline, thereby improving execution time. When an instruction moves from one stage to the next, a new instruction moves into the vacated stage. The Pentium contains two separate pipelines that can operate simultaneously. The first is called the `U-pipe` and the second the `V-pipe.` Most instructions execute in both pipelines in a single clock cycle.

Figure 6.20

Inside the Pentium microprocessor.

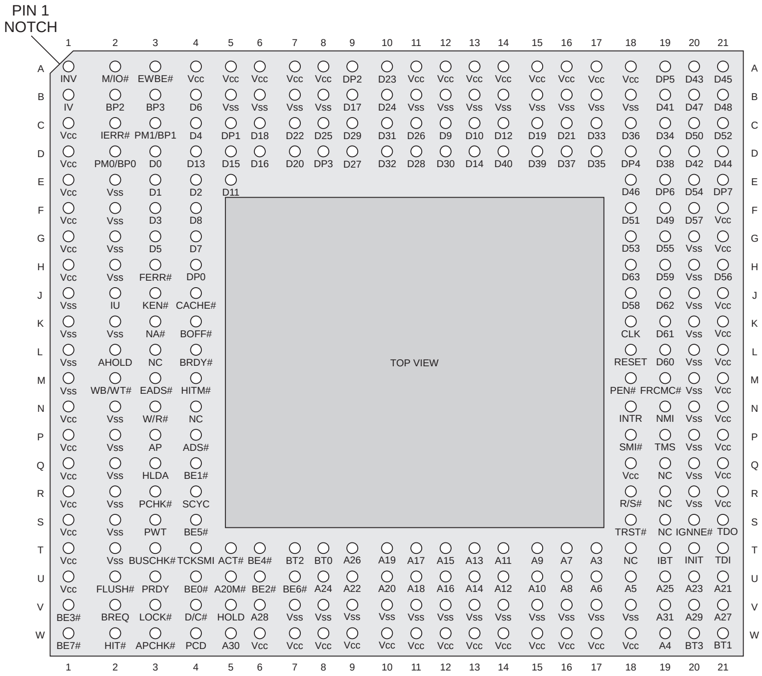


A few pins on the Pentium differ from the 80486. A pin-out for the Pentium is shown in Figure 6.21 and defined in Figure 6.22. The Pentium has been produced in a variety of clock-frequency versions: 75/90/100/120/133/150/166/200 MHz. The external frequency applied to its clock input is the same frequency the

Pentium uses internally. The address bus connections are labeled A3–A31. These lines are bidirectional, as was the case with the 80486, and are used as inputs for cache inquiries. D0–D63 are the 64-bit, bidirectional data bus pins.

Figure 6.21

A pin-out for the Pentium.



Advanced Pentium Architectures

Intel has advanced its Pentium line of microprocessors by introducing three new specifications. These are the Pentium MMX, Pentium Pro, and Pentium II processors.

Pentium MMX

Objective

In the Pentium MMX processor, the multimedia and communications processing capabilities of the Pentium device is extended by the addition of 57 multimedia-specific instruction to the instruction set. Intel also increased the onboard L1 cache size to 32KB. The cache has been divided into two separate 16KB caches: the instruction cache and the data cache. The external L2 cache used with the MMX is typically 256 or 512KB.

Figure 6.22

The pins of the Pentium microprocessor.

ADDRESS									
A3	AL35	A9	AK30	A15	AK26	A21	AF34	A27	AG33
A4	AM34	A10	AN31	A16	AL25	A22	AH36	A28	AK36
A5	AK32	A11	AL31	A17	AK24	A23	AE33	A29	AK34
A6	AN33	A12	AL29	A18	AL23	A24	AG35	A30	AM36
A7	AL33	A13	AK28	A19	AK22	A25	AJ35	A31	AJ33
A8	AM32	A14	AL27	A20	AL21	A26	AH34		

DATA									
D0	K34	D13	B34	D26	D24	D39	D10	D52	E03
D1	G35	D14	C33	D27	C21	D40	D08	D53	G05
D2	J35	D15	A35	D28	D22	D41	A05	D54	E01
D3	G33	D16	B32	D29	C19	D42	E09	D55	G03
D4	F36	D17	C31	D30	D20	D43	B04	D56	H04
D5	F34	D18	A33	D31	C17	D44	D06	D57	J03
D6	E35	D19	D28	D32	C15	D45	C05	D58	J05
D7	E33	D20	B30	D33	D16	D46	E07	D59	K04
D8	D34	D21	C29	D34	C13	D47	C03	D60	L05
D9	C37	D22	A31	D35	D14	D48	D04	D61	L03
D10	C35	D23	D26	D36	C11	D49	E05	D62	M04
D11	B36	D24	C27	D37	D12	D50	D02	D63	N03
D12	D32	D25	C23	D38	C09	D51	F04		

CONTROL							
A20M#	AK08	BREQ	AJ01	HIT#	AK06	PRDY	AC05
ADS#	AJ05	BUSCHK#	AL07	HITM#	AL05	PWT	AL03
ADSC#	AM02	CACHE#	U03	HLDA	AJ03	R/S#	AC35
AHOLD	V04	CPUTYP	Q35	HOLD	AB04	RESET	AK20
AP	AK02	D/C#	AK04	IERR#	P04	SCYC	AL17
APCHK#	AE05	D/P#	AE35	IGNNE#	AA35	SMI#	AB34
BE0#	AL09	DP0	D36	INIT	AA33	SMIACT#	AG03
BE1#	AK10	DP1	D30	INTR/LINT0	AD34	TCK	M34
BE2#	AL11	DP2	C25	INV	U05	TDI	N35
BE3#	AK12	DP3	D18	KEN#	W05	TDO	N33
BE4#	AL13	DP4	C07	LOCK#	AH04	TMS	P34
BE5#	AK14	DP5	F06	M/IO#	T04	TRST#	Q33
BE6#	AL15	DP6	F02	NA#	Y05	VCC2DET#	AL01
BE7#	AK16	DP7	N05	NMI/LINT1	AC33	W/R#	AM06
BOFF#	Z04	EADS#	AM04	PCD	AG05	WB/WT#	AA05
BP2	S03	EWBE#	W03	PCHK#	AF04		
BP3	S05	FERR#	Q05	PEN#	Z34		
BRDY#	X04	FLUSH#	AN07	PM0/BP0	Q03		
BRDYC#	Y03	FRCMC#[1]	Y35	PM1/BP1	R04		

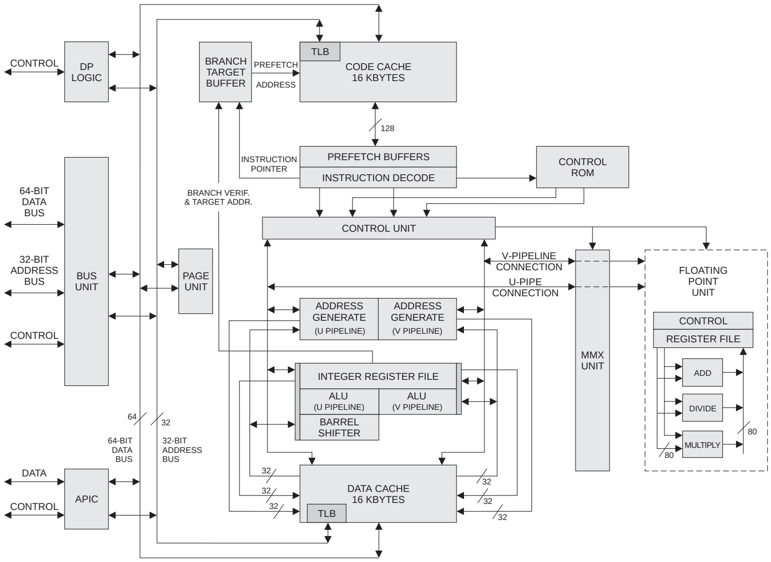
[1] The FRCMC# pin is not defined for the Pentium® processor with MMXTM technology. This pin should be left as a "NC" or tied to VCC3 via an external pullup resistor on the Pentium processor with MMX technology.

The MMX adds an additional multimedia-specific stage to the integer pipeline. This integrated stage handles MMX and integer instructions quickly. Improved branching prediction circuitry has also been implemented to offer higher prediction accuracy and, thereby, provide higher processing speeds. The MMX's four prefetch buffers can hold up to four successive streams of code. The four write buffers are shared between the two pipelines to improve the MMX's memory-write performance. The block diagram of the Pentium MMX is presented in Figure 6.23.

The Pentium MMX processor is available in 166- and 200-MHz versions and is pin-for-pin compatible with previous Pentium processors (273-pin PGA format). However, it requires two separate operating voltages. One source is used to drive the Pentium processor core; the other is used to power the processor's I/O pins. The pin-out of the Pentium MMX is identical to that of the basic Pentium described in Figures 6.22 and 6.23.

Figure 6.23

Inside the Pentium MMX.



Pentium Pro

Intel departed from just increasing the speed of its Pentium processor line by introducing the Pentium Pro Processor. Although compatible with all the previous software written for the Intel processor line, the Pentium Pro is optimized to run 32-bit software. However, it did not remain pin-compatible with the previous Pentium processors. Instead, Intel adopted a 2.46×2.66-inch, 387-pin PGA configuration to house a Pentium Pro processor core, and an onboard 512 or 256KB L2 cache. The L2 cache complements the 16KB L1 cache in the Pentium core. This arrangement is illustrated in Figure 6.24. Notice that although on the same PGA device, the two components are not integrated into the same IC. The unit is covered by a gold-plated, copper/tungsten heat spreader.

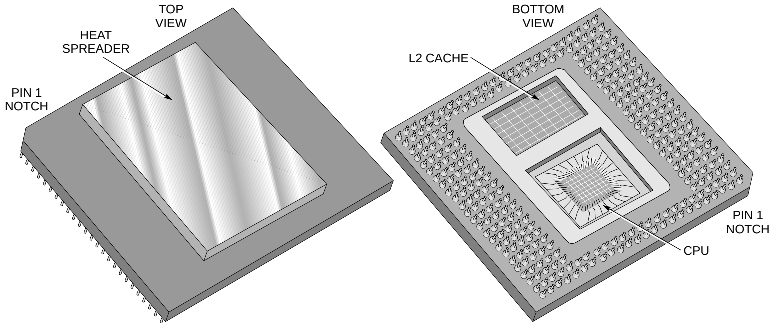
The Pentium Pro uses dynamic execution techniques to manipulate the data flow through it, instead of just processing it. The

Pentium Pro performs the following three types of special data manipulation:

- Multiple branch prediction
- Data flow analysis
- Speculative execution

Figure 6.24

The Pentium Pro microprocessor.



A multiple branch prediction algorithm allows the Pentium Pro to anticipate branching in the instruction sequence. It predicts where the next instruction address will be located in memory by looking ahead in the instruction queue. As the processor decodes instructions, its data-flow-analysis circuitry determines whether the instruction can be executed right away, or whether it depends on the outcome of other instructions. This technique allows the processor to execute the instruction flow in the most efficient manner. In addition, the Pentium Pro processor uses the look-ahead function to process up to five instructions in the pipeline on a speculative basis. When the final state of the instruction sequence is available, the instructions are returned to their proper order and the final results can be output.

The level-2 (L2) onboard cache stores the most frequently used data not found in the processor's internal L1 cache, as close to the processor core as it can be without being integrated directly into the IC. A high-bandwidth cache bus connects the processor and cache unit together. The bus (0.5 inch in length) allows the processor and external cache to communicate at a rate of 1.2GB per second.

The Pentium Pro is designed in a manner so that it can be used in typical, single-microprocessor applications or in multiple-processor environments, such as high-speed, high-volume file servers and work stations. Several dual-processor system boards have been designed for twin Pentium Pro processors. These boards, like the one described in Figure 6.25, are created with two Pentium Pro sockets so that they can operate either with a single processor or with dual processors. When dual processors are installed, logic circuitry in the Pentium Pro's core manages the requests for access to the system's memory and 64-bit buses.

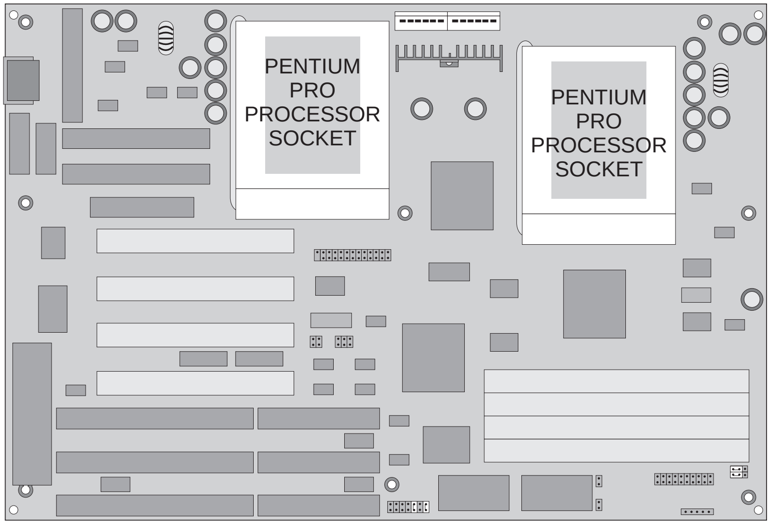


Figure 6.25

A dual-processor system board.

Pentium II

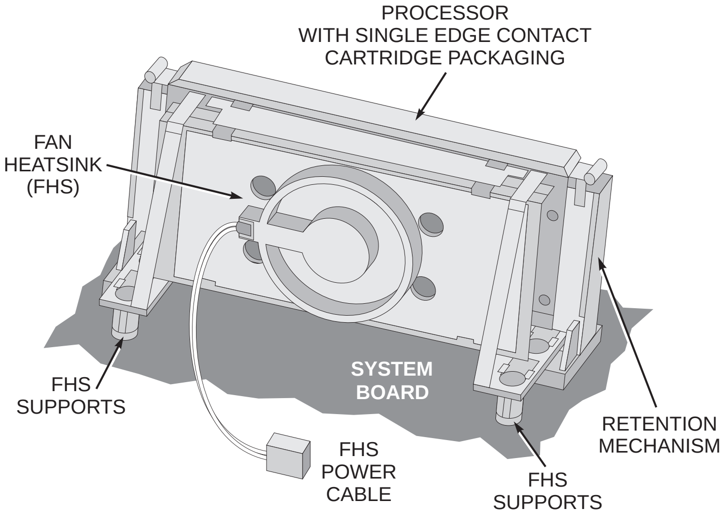
Intel radically changed the form factor of the Pentium processors by housing the Pentium II processor in a new Single Edge Contact (SEC) cartridge, depicted in Figure 6.26. This cartridge uses a special retention mechanism built in to the system board to hold the device in place. The new proprietary socket design is referred to as the `Slot 1` specification and is designed to allow the microprocessor to eventually operate at bus speeds in excess of 300MHz. This is the upper operating frequency limit for pin grid sockets.

The cartridge also requires a special Fan Heat Sink (FHS) module and fan. Like the SEC cartridge, the FHS module requires special support mechanisms to hold it in place. The fan draws power from a special power connector on the system board, or from one of the system's options power connectors.

Inside the cartridge, there is a substrate material on which the processor and related components are mounted. The components consist of the Pentium II processor core, a TAG RAM, and an L2 Burst SRAM. The Pentium II includes all the multimedia enhancements from the MMX processor, as well as retaining the power of the Pentium Pro's dynamic execution and 512KB L2 cache features. The L1 cache is increased to 32KB, while the L2 cache operates with a half-speed bus. Figure 6.27 depicts the Pentium II's cartridge contents.

Figure 6.26

The Pentium II cartridge.



The operation of Pentium Pro and Pentium II processors can be modified by uploading processor update information into BIOS that have application programming interfaces (API) capabilities built in to them. The microprocessor manufacturer places update information on its Web site that can be downloaded to a floppy disk by customers. The user transfers the update information from the update disk to the system's BIOS via the API. If the updated data is relevant (as indicated by checking its processor stepping code), the API writes the updated microcode into the BIOS.

This information will, in turn, be loaded into the processor each time the system is booted. Table 6.1 summarizes the characteristics of the Intel microprocessors covered in this chapter.

Figure 6.27

Inside the Pentium II cartridge.

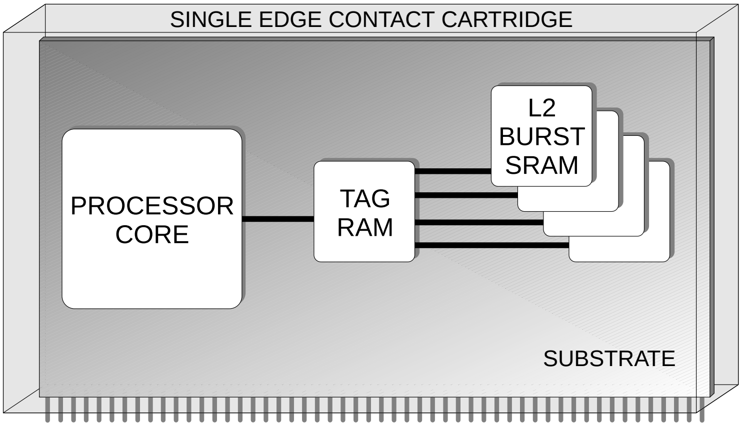


Table 6.1

Characteristics of Intel microprocessors.

Type	Address Bus Width	Space	Internal Clock Speed (MHz)	Data Bus Width	Math Co-Processor	Main Use
8088	20	1MB	4.77(Normal) 8 (Turbo)	8 (supports16)	8087	XT
80286	20 24 24	1MB (real mode) 16MB (protected mode) 16MB (protected mode, addressing & GB of virtual memory)	12	16	80287	AT
80386(DX)	32	4GB	25	32	80387	AT-compat
80386SX	24	16MB(Physical)	16	16-external 32-internal	80387SX	AT-compat
80386SL	32	4GB	20	16-external 32-internal	80387SL	Notebooks
AM386DX	32	4GB	25	32	AM387DX	AT-compat
AM386SX	32	4GB	25	16	AM387SX	AT-compat
IBM386SLC	24	16MB	20	16		PS/2
80486	32	4GB	25	32-external 64-internal	NA	AT-compat

continues

Table 6.1 Continued

Type	Address Bus Width	Space	Internal Clock Speed (MHz)	Data Bus Width	Math Co-Processor	Main Use
80486SX	32	4GB	20	32	80487SX	AT-compat Notebooks
80486DX2	32	4GB	50	32	NA	AT-compat
80486DX4	32	4GB	100	32	NA	AT-compat
CX486SLC	24	16MB	20	13	CX487SLC	Notebooks
CX486DLC	32	4GB	25	32(internal)	NA	AT-compat
CX486DRU2	32	4GB	50	32	NA	AT-compat
IBM486SLC2	24	16 MB	50	16	PS/2	
Pentium	32	4GB	60, 66	64	Onboard	AT-compat
Pentium MMX	32	4GB	133, 150, 166	64	Onboard	AT-compat
Pentium Pro	36	4GB×4	150, 166, 180, 200	64	Onboard	AT-compat
Pentium II	36	64GB	233, 266	64	Onboard	AT-compat

Pentium Clones

As mentioned in Chapter 2, Intel abandoned the 80x86 nomenclature in favor of names that could be copyrighted in an effort to distance themselves from the clone microprocessor manufacturers. When this occurred, the other manufacturers largely followed the 80x86 path but moved toward alternate numbering schemes as well.

Table 6.2 shows the relationship between the various numbering systems. In addition to the 80x86 numbering system, Intel used a Px identification up to the Pentium II. The Pentium II is identified as the Klamath processor. Subsequent improved versions have been dubbed Deschutes, Covington, Mendocino, Katmai, Willamette, Flagstaff (P7), Merced, and Tahoe.

NexGen produced three processors capable of performing at the same level as the P5 (Nx586) and P54C (Nx686) Pentium devices. However, these devices use proprietary pin-outs so that they are not compatible with other processors. Although the performance levels compete with the Pentium, the devices offer compatibility only with 80386/87 operation.

Cyrix uses an Mx numbering system in addition to the 5x/6x86 numbers. The M5/M6/M7 devices are compatible with their Intel counterparts in performance, compatibility, and pin-out. The 5x86 device is compatible with the 80486DX4 in performance, compatibility, and pin-out. The M1 (6x86) and M2 (6x86MX) processors are compatible with the Intel P54C and P55C units in performance and pin-out. The M1 unit is operationally compatible with the 80486DX4; the M2 processor is compatible with the Pentium MMX and Pro processors.

AMD offers five clone microprocessors. These are the 5x86 (X5), the 5x86 (K5), the K6, the K6PLUS-3D, and K7 microprocessors. The X5 offers operational and pin-out compatibility with the DX4. Its performance is equal to the Pentium and MMX processors. The K5 processor is compatible with the Pentium, and the K6 is compatible with the MMX. The K6PLUS-3D is operationally and performance compatible with the Pentium Pro and the K7 is operationally and performance compatible with the Pentium II. Neither of these units, however, has a pin-out compatibility with another processor.

In addition to the previously described clones, Intel has developed a line of upgrade microprocessors for their original units. These are referred to as *OverDrive* processors. The OverDrive unit may simply be the same type of microprocessor running at a higher clock speed, or it may be an advanced architecture microprocessor designed to operate from the same socket/pin configuration as the original. To accommodate this option, Intel has created specifications for 8 socket designs, designated Socket-1 through Socket-8.

The specifications for Socket-1 through Socket-3 were developed for 80486SX, 80486DX, and OverDrive versions that use different pin numbers and power supply requirements. Likewise, Socket-4 through Socket-6 deal with various Pentium and OverDrive units that use different speeds and power supply requirements. The Socket-7 design works with the fastest Pentium units and includes provision for a *Voltage Regulator Module* (VRM) to allow various power settings to be implemented through the socket. Finally, the Socket-8 specification is specific to the Pentium Pro processor.

Table 6.2

Clone processors.

Intel	Cyrix	-AMD	NextGen	Centaur
80386SX (P9)				
80486DX (P4)	M6			
80486SX (P4/P23)	M5			
80486DX2 (P24)	M7			
80486DX4 (P24C)	M1SE (5X86)	-X5 (5X86)		
Pentium (P5/P54C)	M1 (6X86)	-K5 (5X86)	NX586/ 686	C6
Pentium MMX (P55C)	M2 (686MX)	-K6		C6PLUS
Pentium Pro (P6)	MXi	-K6PLUS-3D		
Pentium II	M3	-K7		

Power Supply Requirements

It should be apparent that there are three compatibility issues to consider when dealing with clone processors. These are performance, operation, and pin-out compatibility. In addition to these three issues, it is important to be aware of the power-supply requirements for the various types of microprocessors. In the quest for higher operating speeds, one method of improving microprocessor performance is to use a lower voltage level. This in effect moves the high- and low-logic levels closer together, requiring less time to switch back and forth between them.

Beginning with the Pentium MMX, Intel adopted dual-voltage supply levels for the overall IC and for its core. Common Intel voltage supplies are +5/+5 for older units, and +3.3/+3.3, +3.3/+2.8, +3.3/+1.8 for newer units. Clone processors may use compatible voltages (especially if they are pin-out compatible), or may use completely different voltage levels. Common voltages for clone microprocessors include +5, +3.3, +2.5, and +2.2. The additional voltage levels are typically generated by special regulator circuits on the system board. In each case, the system board's User's Guide should be consulted anytime the microprocessor is being replaced or upgraded.

Microprocessor Support Systems

As powerful as modern microprocessors are, they still require a certain amount of support circuitry to be considered a system. Over time, these support systems have been reduced to just a few application-specific integrated circuit (ASIC) chips. The system boards displayed in Figures 6.1 and 6.4, as well as most other AT-compatible system boards, employ a single ASIC to perform most of the system's AT-compatible functions. This IC is referred to as an **integrated peripheral controller (IPC) chip**.

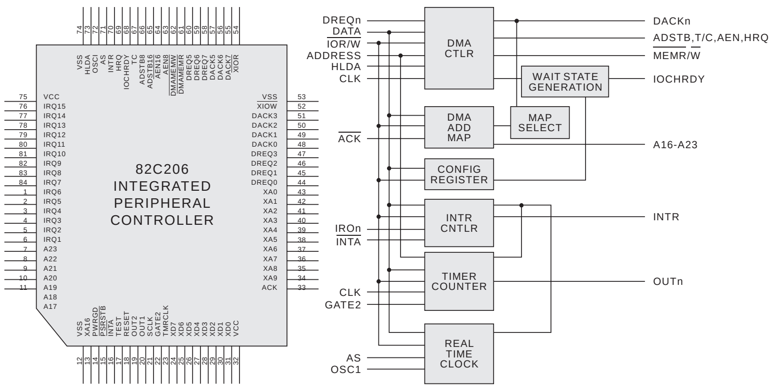
Objective

Integrated Peripheral Controllers (IPC)

The functional blocks of a typical IPC are shown in Figure 6.28. This IC contains the equivalent of the AT's two 8237 four-channel DMA controllers, two 8259 eight-line interrupt controllers, an 8253 three-channel timer/counter IC, and a 74LS612 memory mapper. In addition, the IPC incorporates the system's real-time clock circuit, nonvolatile RAM configuration registers, and all of the AT system's discrete support circuitry for these devices.

Figure 6.28

An integrated peripheral controller (IPC).



Timing Systems

The main clock signal on any system board is the processor clock signal. This signal serves as the reference for all system operations, and is specified in megahertz (MHz). Normally, all of the system board's auxiliary clock signals are derived from this signal.

Typical auxiliary system board frequencies include a 14.318-MHz signal that is applied to the PC-Bus, ISA, and EISA expansion slots to form their oscillator (OSC) line. A 1.19318-MHz signal is also generated to provide the system's peripheral clock (PCLK) signal that drives the system's timer/counter channels.

Most system boards support a software speed switch that can be activated directly from the keyboard to speed up or slow down the operation of the unit. To place the system in turbo speed, press down and hold the Ctrl and Alt keys. While holding these keys, press the plus (+) key on the keyboard's numeric keypad. This operation is written as Ctrl/Alt/+. The system is now running in high-speed turbo mode.

To return the system to normal speed operation, press the Ctrl+Alt+– (minus) key combination. In addition to setting turbo speed operation from the keyboard, the user can select the system's operating speed through a push-button Turbo Switch located on the front panel of the system unit. To return to hardware speed selection, press the Ctrl+Alt+* key combination.

Real-Time Clock

Objective

The Real-Time Clock (RTC) module is included for maintaining the system's time and date. In a PC-compatible system, this time and date information is attached to files when they are written to a disk. This operation is a function of the operating system's file-handling routines. This subsystem contains 114 bytes of RAM information. The IPC's internal clock circuitry uses 14 bytes, and the remainder is used to hold the system's configuration information. This information is backed up by an external, rechargeable battery so that it is retained when the computer is turned off. Table 6.3 shows the addresses and descriptions of the IPC's internal RAM memory locations.

Table 6.3

IPC's internal RAM configuration locations.

Address	Description	Address	Description
00	SECONDS	10	FDD A: AND B: TYPE BYTE
01	SECOND ALARM	12	HDD TYPE BYTE
02	MINUTES	14	EQUIPMENT BYTE

Address	Description	Address	Description
03	MINUTE ALARM	15	LOW BASE MEMORY
04	HOURS	16	HIGH BASE MEMORY
05	HOUR ALARM	17	LOW EXPANSION MEMORY
06	DAY OF WEEK	18	HIGH EXPANSION MEMORY
07	DATE OF MONTH	19	HIGH EXPANSION MEMORY
08	MONTH	1A	DISK C: EXTENDED BYTE
09	YEAR	1B	DISK D: EXTENDED BYTE
0A	STATUS REGISTER A	1C	CMOS CHECKSUM (2 BYTES)
0B	STATUS REGISTER B	1E-2F	LOW EXPANSION MEMORY BYTE
0C	STATUS REGISTER C	30	HIGH EXPANSION MEMORY BYTE
0D	STATUS REGISTER D	31	DATA CENTURY BYTE
0E	DIAGNOSTIC STATUS BYTE	32	INFORMATION FLAGS
0F	STATUS BYTE	34-7E	USER RAM

The first 10 bytes of the RTC hold the time, calendar, and alarm data for the system's software. The DOS date and time commands obtain their values from this table. These bytes are updated once every second. Information is stored in these locations as BCD data.

The alarm bytes can be programmed to generate interrupts at specific times or on a periodic basis. Specifically timed alarm interrupts can be generated by just entering the desired time into the alarm byte registers. Conversely, periodic alarms are generated by programming logic 1s into the two most significant bits of any of the three alarm registers.

The 114 bytes of general-purpose user RAM, located between hex addresses 0E and 7F, are not affected by the RTC update circuitry.

These registers hold the system's CMOS setup configuration information and can be accessed by the system at any time. Because of its nonvolatile nature (battery backup), this memory area is often used to hold these configuration and calibration parameters that must be preserved when the computer is turned off. In Pentium systems, this includes enabling settings for the system board's integrated I/O devices.

Timer/Counter Channels

Objective

The operation of any PC-compatible timer/counter subsystem is identical to the three-channel 8253 timer/counter IC used in PCs and XTs. All three counters are driven from the system's 14.318-MHz `oscillator` (OSC) signal. Counter 0 is connected to the interrupt controller and is used as a general-purpose timer for the system, providing a constant time base for implementing the system's time-of-day clock. Counter 1 may be programmed to produce pulses that initiate dynamic RAM refresh operations. Counter 2 is used to support tone generation for the system's speaker. Table 6.4 lists the system's timer/counter channel definitions.

Table 6.4

System Timer/Counter.	*Channel definitions.*
CHANNEL 0	SYSTEM TIMER
GATE0	TIED ON (INTERNALLY)
CLOCK	1.19318 MHz TMRCLK
OUT0	DRIVES IRQ0 (INTERNALLY) WITH 15 µSEC PULSE
CHANNEL 1	REFRESH REQUEST GENERATOR
GATE1	TIED ON (INTERNALLY)
CLOCK	1.19318 MHz TMRCLK
OUT1	REQUEST REFRESH CYCLE
CHANNEL 2	TONE GENERATOR FOR SPEAKER
GATE2	PORT-B BITS 0 AND 1 (HEX ADDRESS 61)
CLOCK	1.19318 MHz TMRCLK
OUT2	DRIVES SPEAKER

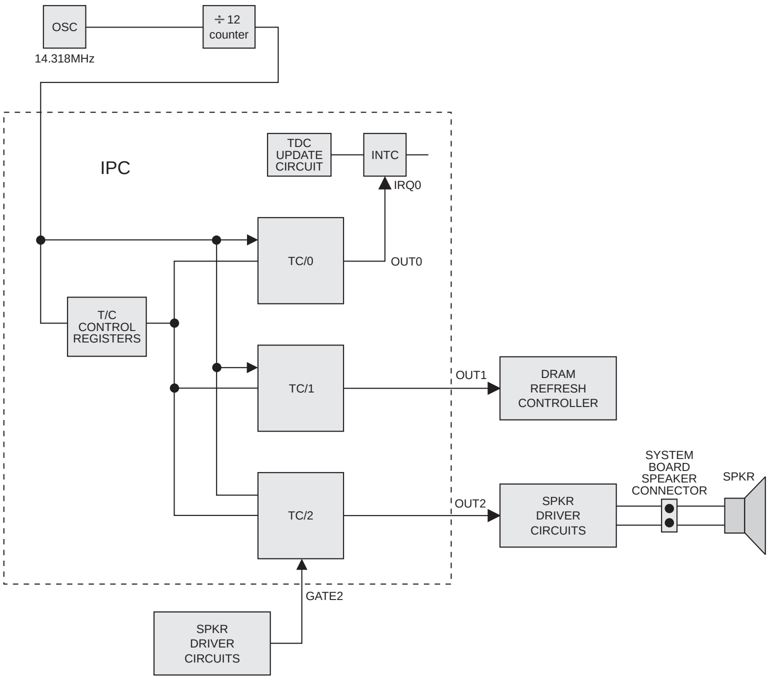
The internal timer/counter circuitry is depicted in Figure 6.29. Each counter can be programmed individually by writing control words into the IPC's control register at hex address locations

040h–043h. The control logic circuitry decodes the control information and provides the internal signals necessary to load, configure, and control each counter.

Each of the timer/counter's channels consists of a 16-bit down-counter that can accept an input clock signal ranging from 0 to 2 MHz. The three channels operate independently of one another, and each can be programmed to produce an output frequency equal to its input frequency divided by any whole number between 1 and 65,535.

Figure 6.29

The timer/counter circuits.



All three counters receive clocking through the IPC's `Timer-Clock` (TMRCLK) input signal. This signal originates from an external clock source and is, therefore, independent from all other IPC clock signals. The 14.318-MHz OSC clock signal is applied to a divide-by-12 counter. The counter outputs the 1.19318-MHz (TMRCLK) clock signal.

The timer's OUT0 output is connected directly to the master interrupt controller's IRQ0 input and is used by the system for time-keeping and task-switching functions. The internal interrupt controller uses this signal to drive the system's time-of-day clock update circuitry.

Counter 1 is used to generate pulses that act as a time-base for the system's memory refresh operations. The OUT1 signal is used to provide the 15-microsecond clock pulses that set up the system's dynamic RAM refresh operations.

Objective

The output of Counter 2 is used to drive the system's speaker circuitry. This is the only timer/counter channel that has an external inhibit line (GATE2). To activate the speaker, the speaker circuit must enable counter 2 and the speaker. The entire range of audio frequencies can be produced by the system's speaker circuitry.

Interrupt Controllers

An AT-compatible interrupt controller device provides two eight-line interrupt controllers (INTC1 and INTC2), each of which is equivalent to the 8259 PIC used in the original PCs, XTs, and ATs. These interrupt controllers are internally cascaded together to provide the 16 interrupt channels necessary for AT-compatibility. Like those discrete 8259s used in the original AT, the IPC's controllers must be programmed to operate in cascade mode. INTC1 is located at hex addresses 020 and 021; INTC2 is located at 0A0 and 0A1.

Of the 16 interrupt channels (IRQ0 through IRQ15) available, three are generally used inside the IPC. Therefore, they do not have external IRQ pins. The other 13 IRQ inputs are available to the system for user-definable interrupt functions. As with the PC and XT, each IRQ input is assigned a priority level. IRQ0 is the highest, and IRQ15 is the lowest. The three internally connected channels are as follows:

Channel 0 (IRQ0) Timer/counter interrupt is okay

Channel 2 (IRQ2) Cascaded to INTC2

Channel 8 (IRQ8) Real-time clock interrupt

Table 6.5 shows the designations for the various interrupt levels in the system.

Table 6.5

System interrupt levels.

Interrupt	Description
NMI	PARITY CHECK ERROR
INTC1	
IRQ0	MINUTE ALARM
IRQ1	KEYBOARD BUFFER FULL
IRQ2	CASCADE FROM INTC2
IRQ3	SERIAL PORT 2
IRQ4	SERIAL PORT 1
IRQ5	PARALLEL PORT 2
IRQ6	FDD CONTROLLER
IRQ7	PARALLEL PORT 1
NMI	PARITY CHECK ERROR
INTC2	
IRQ8	REAL TIME CLOCK
IRQ9	CASCADE TO INTC1
IRQ10	SPARE
IRQ11	SPARE
IRQ12	SPARE
IRQ13	COPROCESSOR
IRQ14	PRIMARY IDE CONTROL
IRQ15	SECOND IDE CONTROL

A typical PC's interrupt circuitry is illustrated in Figure 6.30. Its operation is as follows:

1. When one or more of the interrupt controller's IRQ inputs become active, the interrupt controller checks its internal registers for priority levels.

2. The priority resolver evaluates the priority of the IRQ(s) received, and asserts an INTR signal to the microprocessor.

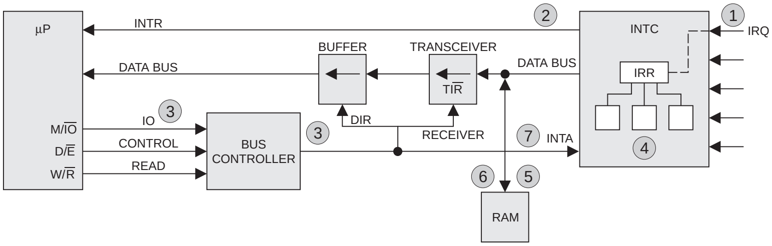


Figure 6.30

The interrupt circuitry.

3. When the microprocessor accepts the interruption, it enters into INTA cycles. These cycles produce the necessary signals (M/IO, D/E, and W/R = 0) to cause the bus controller to issue an INTA signal to the interrupt controller. These conditions also create the proper XDIR and T/R conditions for data movement from the IPC to the microprocessor.

4. Each I/O system must have a special program called an interrupt service routine. This program is specific to that system's function and operational needs. After the controller informs the microprocessor that an interrupt has occurred, it must produce an address to point the microprocessor to the starting address of the service routine that corresponds to the level of interrupt being serviced. This pointing address (vector address) is located in a portion of RAM memory called the vector table. These addresses are physically located in the lowest 1KB (0–3FF) of RAM memory in the system, and contain the starting addresses of the various service routines.

5. The interrupt controller places an 8-bit interrupt vector address (nn) on the XD0–XD7 data pins. The nn value corresponds to the priority of the interrupt being serviced. The microprocessor pushes the contents of its internal registers on a stack and latches the vector address bits during this time.

6. The microprocessor jumps to the address specified by the vector byte and loads a four-byte address. This address

represents the beginning address of the service routine for the interrupting device. The microprocessor services the interrupting device until an `End Of Interrupt` (EOI) instruction is encountered in the routine.

7. The EOI command from the CPU will cause the interrupt controller's ISR bit to be cleared at the end of the second INTA cycle. This marks the end of the service routine. The microprocessor retrieves the contents of its internal registers from the stack and resumes its normal operation at the point where it left off when the interrupt was accepted.

Table 6.6 lists the interrupt vectors and functions used in PC-compatible computer systems. These vectors can also be used by software designers to execute software interrupts in the system. By doing so, they can take over direct control of the system's I/O devices.

Table 6.6

Interrupt vectors.

Interrupt	Function
0	Divide by zero
1	Single step
2	Non-maskable interrupt (NMI)
3	Break point instruction
4	Overflow
5	Print screen
6,7	Reserved
8	Time-of-day hardware interrupt (18.2/sec.)
9	Keyboard hardware interrupt
A	Reserved
B,C	Serial communications hardware interrupts
D	Fixed-disk hardware interrupt
E	Disk hardware interrupt

continues

Table 6.6 Continued

Interrupt	Function
F	Printer hardware interrupt
10	Video I/O call
11	Equipment check call
12	Memory check call
13	Disk I/O call
14	RS232 I/O call
15	(not used)
16	Keyboard I/O call
17	Printer I/O call
18	ROM basic entry code
19	Bootstrap loader
1A	Time of day call
1B	Get control on keyboard break
1C	Get control on timer interrupt
1D	Pointer to video initialization table
1E	Pointer to disk parameter table
1F	Pointer to graphics character generator
20	DOS program terminate
21	DOS function call
22	DOS terminate address
23	DOS CTRL-BRK exit address
24	DOS fatal error vector
25	DOS absolute disk read
26	DOS absolute disk write
27	DOS terminate, fix in storage
28-3F	Reserved for DOS
40-5F	Reserved
60-67	Reserved for user-software interrupts

Interrupt	Function
68-7F	(not used)
80-85	Reserved by BASIC
86-F0	Used by BASIC interpreter while running
F1-FF	(not used)

Objective

Non-Maskable Interrupts (NMI)

Two system board-based conditions will cause a non-maskable interrupt (NMI) signal to be sent to the microprocessor. The first condition occurs when an active `IO Channel Check` (IOCHCK) input is received from an options adapter card located in one of the board's expansion slots. The system's BIOS program enables this signal during initialization. The other event that will cause an NMI signal to be generated is the occurrence of a `Parity Check` (PCK) `error` in the DRAM memory. The BIOS program also enables the parity check signal during the system's initialization.

Objective

DMA Controllers

Before a DMA data transfer begins, the starting address and number of bytes to be transferred are sent to the DMA controller on the data bus, along with information specifying the type and direction of transfer to be performed (I/O-memory, memory-I/O, or memory-to-memory). This information is stored in the DMA controller's internal registers.

During the transfer, the address that the controller applies to the memory unit is changed, or decremented, by 1 each time a byte is transferred. An internal count register, which was originally loaded with the number of bytes to be transferred, is decremented by 1. Once activated, the data transfer continues until the count register has been decremented to 0, or until the I/O device activates the `End Operation` (EOP) line to terminate DMA operations.

The IPC's DMA subsystem provides an AT-compatible PC with four channels for 8-bit DMA transfers (DMA1) and three channels (DMA2) for 16-bit DMA transfers. The DMA1 channels are used to carry out DMA transfers between 8-bit options adapters and

8- or 16-bit memory locations. These 8-bit transfers are conducted in 64 KB blocks and can be performed throughout the system's entire 16-MB address space. The DMA2 channels (channels 5, 6, and 7) are used only with 16-bit devices and can only transfer words in 128-KB blocks. The first 16-bit DMA channel (DMA channel 4) is used internally to cascade the two DMA controllers together. Table 6.7 describes the system's DMA channel designations.

Table 6.7

System's DMA channel designations.

Channel	Function	Controller	Page Register Address
CH0	SPARE	1	0087
CH1	SDLC (NETWORK)	1	0083
CH2	FDD CONTROLLER	1	0082
CH3	SPARE	1	0081
CH4	CASCADE TO CNTR 1	2	
CH5	SPARE	2	008B
CH6	SPARE	2	0089
CH7	SPARE	2	008A

DAM Controller Modes

The typical DMA subsystem operates in three distinctly different modes, as follows:

- Idle Mode
- Program Mode
- Active Mode

When no DMA requests are pending, the DMA subsystem operates in idle mode. In this mode, the DMA controller performs `single state` (S1) operations. This is the DMA controller's default mode and the mode in which it will operate unless a DMA request is presented to the controller, or one of its internal registers is accessed by the system for programming.

When the system places a valid address on the address bus that corresponds to one of the IPC's DMA registers (along with an IOR or IOW signal), the DMA subsystem enters program mode. In this mode, all programmable aspects of the DMA channel can be written to (or read from) the DMA controller.

When the DMA subsystem is presented with a DMA request on one of its DRQ lines, it enters into its active mode. A normal active DMA cycle consists of three states (S2, S3, and S4), which require 4 DMA clock cycles to complete. State S3 is normally executed twice to create a 1 wait-state condition in the DMA timing sequence. However, the DMA channel can be programmed to produce higher data throughput by excluding the second S3 state and terminating the DMA cycle in S4. This would, of course, produce a zero wait-state condition.

A Typical DMA Operation

Figure 6.31 depicts a typical DMA transfer from floppy disk to memory. In this case, the IPC's DMAMEMW, XIOR, and DACK2 output lines are used to move bytes of data from the disk drive controller, through the expansion-slot connector and I/O buses, and into RAM memory. The DMAMEMW line works in conjunction with the address that the DMA controller places on the address bus to select the location in RAM memory where the byte of information from the disk drive controller will be stored.

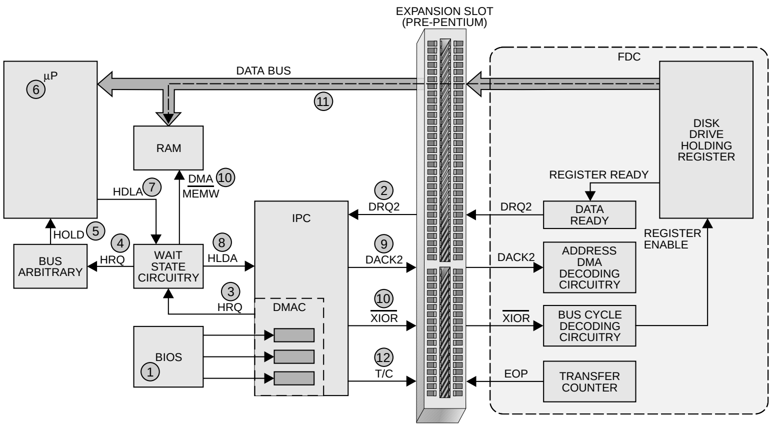


Figure 6.31

A typical DMA operation.

The XIOR line works with the DACK2 line to access and read the disk drive controller's output register. Because the address bus is being used at this time to access the RAM memory location, the DACK line is needed to act as the chip select signal that addresses the output latch on the controller board. Therefore, the DACK line performs the same task that a decoded I/O address would on the controller card. Note that during I/O-to-memory or memory-to-I/O transfers, the DMA controller does not handle any of the data being transferred internally. It just manipulates the I/O device and the memory structure to carry out the data transfer.

The floppy-disk drive DMA transfer operation can be summarized as follows:

1. The IPC's internal registers are loaded with information such as the following:

   a. The address in RAM memory where the data from the FDD will be stored

   b. The number of bytes to be transferred

   c. The size of the block (sector) to be transferred

   d. The mode of transfer (block or byte-by-byte)

2. The FDD controller signals the IPC by activating the system's DRQ2 line.

3. The DMA controller responds by sending a Hold Request (HRQ) signal to the system's wait-state logic.

4. The wait-state circuitry applies the HRQ signal to the bus arbitrator circuitry.

5. The bus arbitrator issues a HOLD signal to the microprocessor.

6. The microprocessor finishes its current instruction and places its address, data, control bus lines in a high-impedance state.

7. The microprocessor generates the Hold Acknowledge (HLDA) signal.

8. The HLDA is applied to the IPC. The HLDA signal is also converted into the Acknowledge (ACK) signal that is applied to the IPC.

9. The IPC activates the DACK2 signal, which is applied to the expansion slot and to the peripheral device (FDD unit) to enable it.

10. The IPC directs the data transfer from the FDC to RAM memory (data does not flow through the controller).

11. As the last byte is transferred, the Terminal Count (T/C) and EOP signals should be generated simultaneously, if the transfer has been carried out successfully.

Memory Systems

Objective

Three types of semiconductor memory are generally found on a typical system board. These include the system's ROM BIOS ICs, the system's RAM memory, and the second-level cache memory unit.

A typical PC system board uses one or two 256KB/128KB × 8 ROM chips to hold the system's basic input/output system (BIOS) firmware. The system's memory map reserves memory locations from F0000h to FFFFFh. These chips contain the firmware routines to handle startup of the system, the change-over to disk-based operations, video and printer output functions, as well as a Power-On Self-Test (POST).

Basically, two types of semiconductor RAM—static RAM (SRAM) and dynamic RAM (DRAM)—are used on system boards. Although they both perform the same types of functions, the methods they use are completely different. SRAM stores bits in such a manner that they will remain as long as power to the chip is not interrupted. DRAM requires periodic refreshing to maintain data, even if electrical power is applied to the chip.

DRAM stores data bits on rows and columns of IC capacitors. Capacitors lose their charge over time. This is the reason that DRAM devices require data-refreshing operations. SRAM uses IC transistors to store data and maintains it as long as power is supplied to the chip. Its transistor structure makes SRAM memory much faster than DRAM. However, it can store only about 25% as much data in a given size as a DRAM device. Therefore, it tends to be more expensive to create large memories with SRAM.

Whether the RAM is made up of static or dynamic RAM devices, all RAM systems have the disadvantage of being volatile. This means that any data stored in RAM will be lost if power to the computer is disrupted for any reason. On the other hand, both types of RAM have the advantage of being fast, with the capability to be written to and read from with equal ease.

Generally, static RAM is used in smaller memory systems, such as cache and video memories, where the added cost of refresh circuitry would increase the cost-per-bit of storage. Cache memory is a special memory structure that works directly with the microprocessor; video memory is a specialized area that holds information to be displayed onscreen. On the other hand, DRAM is used in larger memory systems, such as the system's main memory, where the extra cost of refresh circuitry is distributed over a greater number of bits and is offset by the reduced operating cost associated with DRAM chips.

Memory Overhead

It has already been mentioned that the DRAM devices, commonly used for the system's RAM, require periodic refreshing of their data. Some `refreshing` is performed just by regular reading and writing of the memory by the system. However, additional circuitry must be used to ensure that every bit in the memory is refreshed within the allotted time frame. In addition to the circuitry, the reading and writing times used for refreshing must be taken into account when designing the system.

Another design factor associated with RAM is data error detection. A single, incorrect bit can shut down the entire system instantly. With bits constantly moving in and out of RAM, it is crucial that all the bits be transferred correctly. The most popular form of error detection in PC compatibles is `parity checking`. In this methodology, an extra bit is added to each word in RAM and checked each time it is used. Like refreshing, parity checking requires additional circuitry and memory overhead to operate.

DRAM Refresh

Dynamic RAM devices require that data stored in them be `refreshed`, or rewritten, periodically to keep it from fading away. As a matter of fact, each bit in the DRAM must be refreshed at least once every two milliseconds or the data will dissipate. Because it can't be assumed that each bit in the memory will be accessed during the normal operation of the system (within the time frame allotted), the need to constantly refresh the data in the DRAM requires special circuitry to perform this function.

The extra circuitry and inconvenience associated with refreshing may initially make DRAM memory seem like a distant second choice behind static RAM. Because of the simplicity of DRAM's internal structure, however, the bit-storage capacity of a DRAM chip is much greater than that of a similar static RAM chip, and it offers a much lower rate of power consumption. Both of these factors contribute to making DRAM memory the economical choice in certain RAM memory systems—even in light of the extra circuitry necessary for refreshing.

Parity Checking

Objective

Parity is a simple self-test used to detect RAM read-back errors. When a data byte is being stored in memory, the occurrences of logic "1s" in the byte are added together by the parity generator/checker chip. This chip produces a parity bit that is added to, and stored along with, the data byte. Therefore the data byte becomes a 9-bit word. Whenever the data word is read back from the memory, the parity bit is reapplied to the parity generator and recalculated. The recalculated parity value is then compared to the original parity value stored in memory. If the values do not match, a parity-error condition occurs and an error message is generated. Traditionally, there are two approaches to generating parity bits; the parity bit may be generated so that the total number of 1-bits equals an even number (`even parity`) or an odd number (`odd parity`).

To enable parity checking, an additional ninth bit is added to each byte stored in DRAM. On older systems, an extra memory chip was included with each bank of DRAM. In newer units, the extra storage is built in to the SIMM and DIMM modules. Whether a particular system employs parity check depends on its chip set. Many newer chip sets have moved away from using parity checking altogether. In these cases, SIMMs and DIMMs with parity capability can be used, but the parity function will not function. In Pentium systems, the system board's User's Guide or the BIOS' Extended CMOS Setup screen should be consulted to determine whether parity is supported. If so, the parity function can be enabled through this screen.

The system's parity generator/checker circuitry consisted of discrete 74LS280 ICs in the original PC, PC-XT, PC-AT, and 80386-based compatibles. Figure 6.32 illustrates how the system's RAM and parity-checking circuit work together.

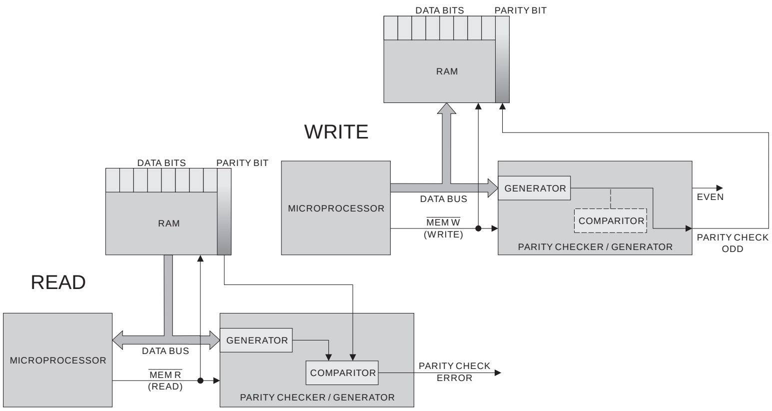


Figure 6.32

How parity checking works.

With the advent of the 80486 processor, the parity-checking function is built directly in to the microprocessor. However, some 486 and Pentium system boards retain the discrete parity generator/checkers for DMA data.

When a parity error occurs, an NMI signal is cogenerated in the system, causing the BIOS to execute its NMI handler routine. This routine will normally place a parity error message onscreen, along with an option to shut down the system or continue.

Advanced Memory Structures

As the operating speeds of microcomputers have continued to increase, it has become increasingly necessary to develop new memory strategies to keep pace with the other parts of the system. Some of these methods, such as developing faster DRAM chips, or including wait states in the memory-access cycles, are very fundamental in nature. The methods do not allow the entire system to operate at its full potential, however. Other, more elaborate memory management schemes have been employed on faster computers to maximize their overall performance.

Cache Memory

Objective

One method of increasing the memory-access speed of a computer is called `caching`. This memory management method assumes that most memory accesses are made within a limited block of addresses. Therefore, if the contents of these addresses are relocated into a special section of high-speed SRAM, the microprocessor could access these locations without requiring any wait states. Cache memory is normally small to keep the cost of the system as low as possible. It is also very fast, however, even in comparison to fast DRAM devices.

Cache memory operations require a great deal of intelligent circuitry to operate and monitor the cache effectively. The cache controller circuitry must monitor the microprocessor's memory-access instructions to determine whether the specified data is stored in the cache. If the information is in the cache, the control circuitry can present it to the microprocessor without incurring any wait states. This is referred to as a `hit`. If the information is not located in the cache, the access is passed on to the system's RAM and it is declared a `miss`.

The primary objective of the cache memory's control system is to maximize the ratio of hits to total accesses (hit rate), so that the majority of memory accesses are performed without wait states. One way to do this is to make the cache memory area as large as possible (thus raising the possibility of the desired information being in the cache). However, the relative cost, energy consumption, and physical size of

SRAM devices works against this technique. Practical sizes for cache memories run between 16–512KB.

There are two basic methods of writing updated information into the cache. The first is to write data into the cache and the main memory at the same time. This is referred to as `write thru cache`. This method tends to be slow because the microprocessor has to wait for the slow DRAM access to be completed. The second method is known as `write back cache`. A write back cache holds the data in the cache until the system has a `quiet` time, and then writes it into the main memory.

The 80486 and Pentium microprocessors have a built-in `first-level cache` that can be used for both instructions and data. The internal cache is divided into four, 2-KB blocks containing 128 sets of 16-byte lines each. Control of the internal cache is handled directly by the microprocessor. The first-level cache is also known as an `L1 cache`. However, many system boards extend the caching capability of the microprocessor by adding an external, `second-level` 256/512-KB memory `cache`. Like the L1 cache, the second level cache may also be referred to as an `L2 cache`. An external L2 cache memory system is depicted in Figure 6.33.

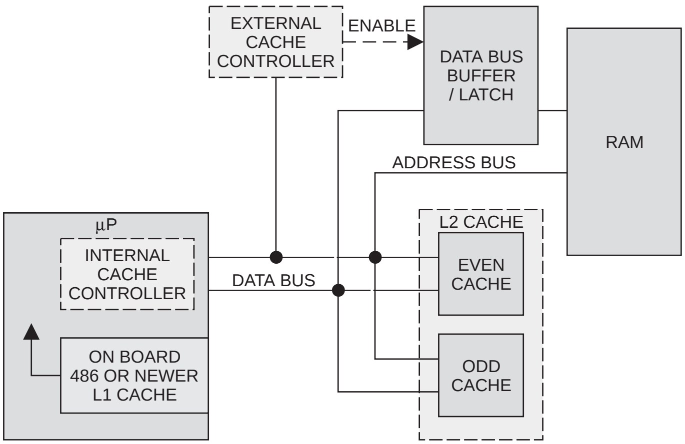


Figure 6.33

An external cache.

Both types of RAM are brought together to create an improved DRAM, referred to as enhanced DRAM (EDRAM). By integrating an SRAM component into a DRAM device, a performance improvement of 40% can be gained. An independent write path allows the system to input new data without affecting the operation of the rest of the chip. These devices are used primarily in L2 cache memories.

SRAM is available in a number of different types:

- Asynchronous SRAM is standard SRAM and delivers data from the memory to the microprocessor and returns it to the cache in one clock cycle.
- Synchronous SRAM uses special buffer storage to deliver data to the CPU in one clock cycle after the first cycle. The first address is stored and retrieves the data while the next address is on its way to the cache.
- Pipeline SRAM uses three clock cycles to fetch the first data and then accesses addresses within the selected page on each clock cycle.
- Burst mode SRAM loads a number of consecutive data locations from the cache, over several clock cycles, based on a single address from the microprocessor.

Memory Paging and Interleaving

There are also other commonly used methods of organizing RAM memory so that it can be accessed more efficiently. Typically, memory accesses occur in two fashions: instruction fetches (which are generally sequential), and operand accesses (which tend to be random). Paging and interleaving memory schemes are designed to take advantage of the sequential nature of instruction fetches from memory.

The basic idea of paged-mode DRAM operations is illustrated in Figure 6.34. Special memory devices called page-mode (or static-column) RAM are required for memory paging structures. In these memory devices, data is organized into groups of rows and columns called pages. After a ROW access is made in the device, it is possible to access other column addresses within the same row without precharging its `Row Address Strobe` (RAS) line. This feature produces access times that are half that of normal DRAM memories. `Fast page mode RAM` is a quicker version of page mode RAM having improved CAS access speed.

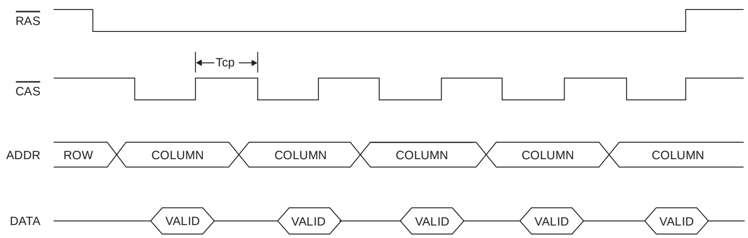


Figure 6.34

Page mode DRAM operation.

The operating principle behind `memory interleaving` is described in Figure 6.35. Typical interleaving schemes divide the memory into two banks of RAM, with one bank storing even addresses and the other storing odd addresses. The RAS signals of the two banks overlap so that the time required to precharge one bank's `RAS` line is used for the active RAS time of the other bank. Therefore, there should never be a precharge time for either bank so long as the accesses continue to be sequential. If a nonsequential access occurs, a miss is encountered and a wait state must be inserted in the timing. If the memory is organized into two banks, the operation is referred to as two-way interleaving. It is also common to organize the memory into four, equal-sized banks. This organization effectively doubles the average `0`-wait state `hit` space in the memory.

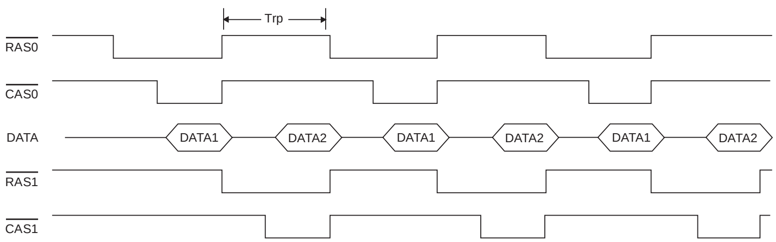


Figure 6.35

Memory interleaving.

Other RAM Types

Another modified DRAM, referred to as `synchronous DRAM` (SDRAM), uses special internal registers and clocks to organize data requests from memory. This allows the microprocessor to perform other tasks while the data is being organized.

`Extended Data Out` (EDO) memory increases the speed at which RAM operations are conducted by cutting out the 10-nanosecond wait time normally required between issuing memory addresses. This is accomplished by not disabling the data bus pins between bus cycles. EDO is an advanced type of fast page mode RAM also referred to as `hyper page mode RAM`. The advantage of EDO RAM is encountered when multiple sequential memory accesses are performed. By not turning off the data pin, each successive access after the first access is accomplished in two clock cycles rather than three.

Special memory devices have also been designed to optimize video-memory–related activities. Among these are `video RAM` (VRAM) and `windows RAM` (WRAM). In typical DRAM devices, access to the data stored inside is shared between the system microprocessor and the video controller. The microprocessor accesses the RAM to update the data in it and to keep it refreshed. The video controller moves data out of the memory to become screen information. Normally, both devices must access the data through the same data bus. VRAM employs a special `dual port access` system to speed up video operations. WRAM is a special version of VRAM optimized to transfer blocks of data at a time. This allows it to operate at speeds of up to 150% of typical VRAM and costs up to 20% less.

Onboard I/O

When dealing with a PC-compatible, there are two forms of I/O to contend with. These include the system board's onboard I/O, and peripheral devices that interact with the system through its expansion slots. In a PC-compatible system, some system I/O addresses are associated with intelligent devices on the system board, such as the interrupt and DMA controllers, timer/counter channels, and keyboard controller. Other system I/O ports and their interfaces are located on optional plug-in cards. These easily installed options give the system a high degree of flexibility in adapting to a wide variety of peripheral devices.

Most of the I/O functions associated with PC-compatible systems have become so standardized that IC manufacturers produce them in single-chip ASIC formats.

Certain I/O connections have become standards associated with PC-compatibles. These include the system's parallel printer ports: RS-232 serial ports and the game port.

In both cases, the I/O controllers integrated into the ASIC are responsible for matching signal levels and protocols between the system and the I/O device.

I/O Addressing

Each I/O device must have its own specific address. Computers can use two common methods to handle I/O addressing. In some computers, the microprocessor addresses I/O in the same manner as it does memory locations. This is because the I/O devices are granted a portion of the available address codes, and the same control signals are used to read and write both I/O and memory locations. This method of I/O addressing is referred to as `memory-mapped I/O`.

In the second method, separate microprocessor control signals and address decoders are used for I/O addressing. Microprocessors in these systems use separate outputs such as `memory request` (MREQ) and `I/O request` (IORQ) to distinguish between memory and I/O operations. This practice of distinguishing between memory and I/O addressing is the more accepted method of handling I/O. It is referred to as `direct I/O` or `isolated I/O`.

There are advantages and disadvantages to both addressing methods. In the memory-mapped method, the same instructions used to reference memory can also be used to move data to or from an I/O device. But, the I/O devices use up a portion of the available address codes, limiting the actual amount of storage available in the memory. In direct I/O systems, none of the memory allocations are used up by I/O devices, but extra control lines are required.

Address Decoding

As the microprocessor's address bus runs through the system and connects the various components together, the binary information on the bus is decoded into a single signal. This decoded signal is used to access the one location in the system where a data READ or WRITE is to be performed.

In IBM-compatible systems, the microprocessor differentiates between memory and I/O accesses because these activities are specified by different instructions in the instruction set. Data transfers between the microprocessor and memory locations, or between the microprocessor's internal registers, are called for by program code that takes the form of an assembly language `move instruction` (MOV), as follows:

> MOV (`Destination memory location`), (`Source memory location`) Input and output operations are carried out between the microprocessor and an I/O port, when an IN or OUT assembly language instruction is encountered.
>
> IN (`Microprocessor register`), (`I/O Port Location`) OUT (`I/O Port Location`), (`Microprocessor register`) The microprocessor responds to these different types of instructions by applying an appropriate logic level to its M/IO pin. The microprocessor holds the M/IO pin active for the MOV instruction, and deactivates it when an IN or an OUT instruction is executed.

The system treats its onboard intelligent devices as I/O addresses. The onboard address decoder, like the one displayed in Figure 6.36, converts address bits from the address bus into chip-select enabling bits for the system's intelligent devices. These addresses are included in the overall I/O addressing map of the system.

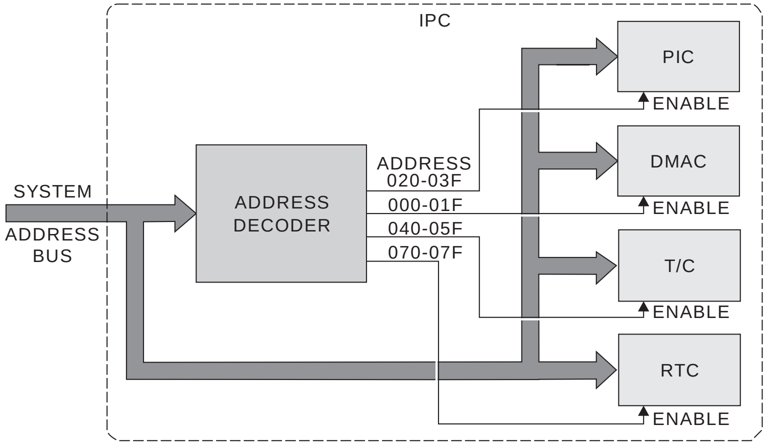


Figure 6.36

Address decoding.

The various I/O port addresses listed in Table 6.8 are used by standard I/O adapters in the PC-compatible system. Notice that these addresses are redundant with those stated for the system's interrupt vectors given in Table 6.9. This method of addressing is referred to as `redundant addressing`. Figure 6.37 illustrates how a system address is routed through the system to an I/O port location.

Table 6.8

I/O port addresses.

Hex Address	Device	Usage
000-01F	DMA CONTROLLER (IPC)	SYSTEM
020-03F	INTERRUPT CONTROLLER (IPC)	SYSTEM
040-05F	TIMER/COUNTER (IPC)	SYSTEM
060-06F	KEYBOARD CONTROLLER	SYSTEM
070-07F	REAL-TIME CLOCK, NMI MASK (IPC)	SYSTEM
080-09F	DMA PAGE REGISTER (IPC)	SYSTEM
0A0-0BF	INTERRUPT CONTROLLER (IPC)	SYSTEM
0F0	CLEAR MATH COPROCESSOR BUSY	SYSTEM
0F1	RESET MATH COPROCESSOR	SYSTEM
0F8-0FF	MATH COPROCESSOR	SYSTEM

Hex Address	Device	Usage
1F0-1F8	HARD-DISK CONTROLLER	I/O
200-207	GAME PORT	I/O
278-27F	PARALLEL PRINTER PORT #2	I/O
2F8-2FF	SERIAL PORT #2	I/O
378-37F	PARALLEL PRINTER PORT #1	I/O
3B0-3BF	MGA/FIRST PRINTER PORT	I/O
3D0-3DF	CGA	I/O
3F0-3F7	FDD CONTROLLER	I/O
3F8-3FF	SERIAL PORT #1	I/O

Table 6.9 System memory map.

Address	Function
0-3FF	Interrupt vectors
400-47F	ROM-BIOS RAM
480-5FF	BASIC and special system function RAM
600-9FFFF	Program memory
0A0000-0AFFFF	VGA/EGA display memory
0B0000-0B0FFF	Monochrome display adapter memory
0B8000-0BFFFF	Color graphics adapter memory
0C0000-0CFFFF	Hard disk ROM
0D0000-0D7FFF	Spare ROM
0D8000-0DFFFF	Spare ROM
0E0000-0E7FFF	Spare ROM
0E8000-0EFFFF	Spare ROM
0F0000-0F3FFF	Spare ROM
0F4000-0F7FFF	Spare ROM
0F8000-0FBFFF	Spare ROM
0FC000-0FDFFF	ROM BIOS
0FE000-0FFFFF	ROM BIOS

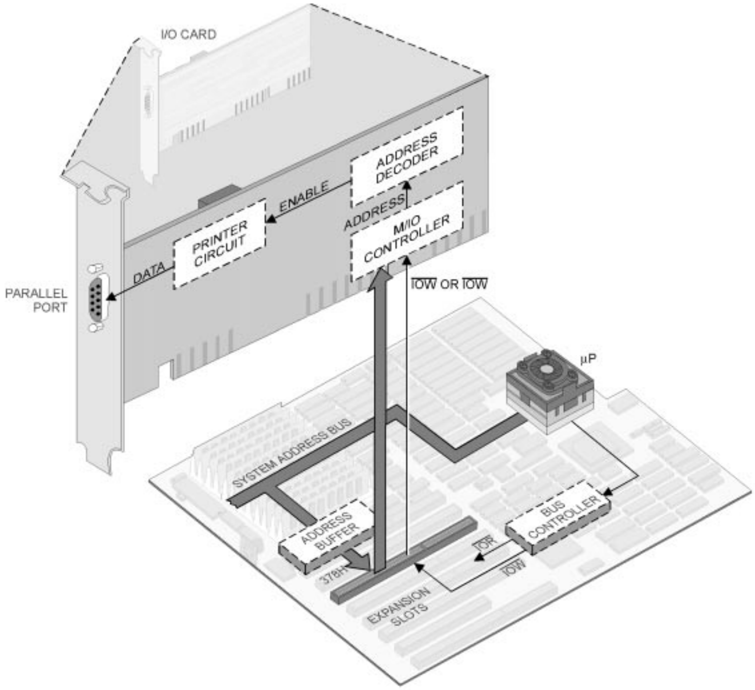


Figure 6.37

Address routing to an I/O port.

Expansion Slots

The system's expansion slots provide the connecting point for most of its I/O devices. Interface cards communicate with the system through the extended microprocessor buses in these slots. As mentioned in Chapter 1, expansion slots basically come in three formats: 8-bit, 16-bit, and 32-bit data buses. The PC-Bus slot is the most famous example of an 8-bit expansion slot; the ISA slot is the consummate 16-bit expansion bus. The 32-bit expansion buses include the MCA bus, the EISA bus, the VESA bus, and the PCI bus.

PC-Bus Expansion Slot

This expansion bus provided the I/O channel in the original PC, PC-XT, and their compatibles. It became the de facto connection standard in the industry for 8-bit systems.

ISA Expansion Slots

The 16-bit Industry Standard Architecture (ISA) slot is the most common expansion slot used with microcomputers. This bus specification originally appeared on the 16-bit, 80286-based PC-AT system board. At that time it was called the AT bus. However, its widespread acceptance earned it the ISA title it now carries. Even in units that have newer, faster 32-bit expansion slots, it is not uncommon to find one or more ISA slots.

Notice that although the physical aspects are identical, some of the pins of the 62-pin expansion slot have been redefined from the PC-Bus. In particular, pins SLB4 and SLB19 have been changed from IRQ2 and DACK0 to IRQ9 and REFRESH, respectively. The definition of SLB8 has changed from `Reserved` to `0WS`, a line used with memory devices to allow 0 wait-state operations. Finally, many of the address, data, and control lines have had an S (for System) identifier added to their definitions.

One of the most interesting features of the 36-pin auxiliary slot is the presence of a second pair of READ/WRITE enable lines: MEMR and MEMW. These lines serve memory locations associated with the highest four address bits (LA20–LA23). Addresses in the lowest megabyte of memory are served by the SMEMR and SMEMW lines in the 62-pin slot.

On many AT-compatible system boards, it is common to find that the designer has retained two or three of the 62-pin, 8-bit PC-Bus expansion slots. Only options adapter cards with a PC-Bus–compatible, 8-bit, edge connector can be used in these slots. This does not mean, however, that these 8-bit cards must be installed in one of the 8-bit slots. They will function just as well if they are placed in the 62-pin portion of an ISA-compatible 16-bit slot.

Advanced 32-Bit Architectures

Objective

As 32-bit microprocessors have gained in popularity, the shortcomings and restrictions of the 16-bit ISA bus have become more noticeable. As already noted, the ISA bus cannot support the full, 32-bit capabilities of microprocessors such as the 80386DX and i486. In addition, the physical organization of the signal lines in the ISA bus produce unacceptable levels of radio frequency interference (RFI) as the bus speed increases.

When the 80386DX first appeared, some manufacturers began producing computers that reserved one special, 32-bit expansion slot for proprietary I/O cards. The 32-bit slot is usually constructed by adding a second 62-pin connector to the standard 8-bit PC-Bus connector. The signals in the first portion of the connector are normally identical to those of the normal PC-Bus slots. The manufacturer adds all the new address, data, and control signals for the 32-bit function to the second 62-pin connector. All the other expansion slots remained ISA compatible.

Most of the I/O boards produced for these slots are large memory cards that allow the system's microprocessor and memory to operate together (at very high speeds), separately from the system board's I/O functions. The slot effectively provides a 32-bit memory bus structure that works in parallel with the system's main bus. This technique of separating the buses is called bifurcation.

The main problem with this approach to achieving a 32-bit expansion bus is the lack of compatibility that results. Because the memory boards are proprietary, it is not usually possible to use one manufacturer's memory card in another manufacturer's system board. In addition, a large part of what personal computers do involves accessing disks, video output systems, and other I/O devices. Therefore, the system will still be prevented from performing at its maximum potential much of the time.

Two legitimate 32-bit bus standards have been developed to take advantage of the full power of the 32-bit microprocessors. These are the Expanded Industry Standard Architecture (EISA) bus, which, as its name implies, is an extension of the existing ISA standard bus, and a new IBM-sponsored bus standard called Micro Channel Architecture (MCA).

EISA Systems

The EISA bus specification naturally adds 16 more data lines and 16 additional address lines to the ISA bus. In the EISA design, all

32 address lines on the bus are latched, so they will hold the address information throughout the complete address cycle. Several of the control lines associated with more powerful microprocessors (such as the 80386DX and 80486) are extended directly to the EISA bus. These lines allow the system to transfer 8-, 16-, or 32-bit words.

Conversely, the standard also includes lines to let I/O cards specify which type of transfer they are capable of. These are the EX16 and EX32 lines. If neither line is activated during a data transfer, the EISA system assumes that an 8-bit transfer is being performed. The EISA standard calls for systems to incorporate bus controller circuitry that will automatically dissect larger data words into sizes manageable by the I/O card.

To retain compatibility with the older ISA options adapters, the clock speed of the bus must be restricted to between 6MHz and 10MHz in EISA systems. To overcome the relative slowness of the bus, the EISA standard provides two high-speed data transfer options.

First, the EISA standard makes provisions to allow full `burst-mode DMA` transfers to occur. In typical PC architectures, DMA transfers are carried out one word at a time. However, the EISA structure allows the DMA controller to seize the bus and conduct continuous transfers.

The second high-speed data transfer method allowed in the EISA specification is called `compressed transfer.` Using the compressed method causes the data to be moved twice as quickly across the bus as it normally would be.

Bus Mastering

The point where the EISA standard departs from previous microcomputer architecture is found in its bus-sharing capabilities. The microcomputer systems you have studied thus far were designed on the premise that a single microprocessor (and maybe a math coprocessor) would be working on a single task. More powerful architectures, like those of mainframe and minicomputers, however, allow multiple `bus masters` (microprocessors and intelligent controller devices) to access the bus and perform multitasking and parallel-processing applications. In these applications, it is possible for different processors within the system to divide the workload and attack different parts of a task simultaneously.

In these systems, options are divided into two categories: bus masters (those devices that can take control of the system's buses) and `slaves` (those devices that must be controlled by some other intelligent device). The EISA standard allows for up to seven bus masters (the system's microprocessor and six bus masters) to operate within the system at any given time.

Bus-mastering systems require special circuitry to act as a referee between all the potential bus masters within the system. This circuitry arbitrates between different devices that request access to the system buses at the same time, and decides which device should be given access. Under the EISA standard, arbitration levels are assigned to devices in a fixed, three-tiered arrangement.

The highest level of access is given to the DRAM refresh circuitry, the second level of access is assigned to DMA operations, and the third level is shared between the system microprocessor and the other six possible bus masters. Different levels of priority are assigned to the different members of tiers two and three. If several channels are requesting service when control is passed to the DMA phase of the cycle, for example, only the DMA channel with the highest priority assignment will be granted access to the buses at that time. Within the DMA channels, priority levels are assigned on a rotating basis. As the priority level is shifted around the upper tier of DMA channels, the cascade position is eventually reached. When this occurs, the DMA channel having the highest priority in the lower tier will be granted access to the buses. After the DMA operation is completed, control shifts to the microprocessor/bus-master tier. Like the DMA tier, control in this tier passes back and forth between the microprocessor and bus-master positions. When control passes to the bus-master position, the bus master with the highest priority level is granted access to the buses.

The bus-master arbitration function is designed into a special VLSI bus controller device that contains the EISA-compatible bus control functions. Because hardware, rather than software, is used to control bus access, this action is referred to as `hardware-mediated bus arbitration`. Recall that it is possible to perform multitasking operations using a 286-based PC. In a typical ISA architecture, however, software is responsible for deciding when and how tasks are operated on. Therefore, multitasking in these computers is referred to as `software-mediated bus arbitration`.

When an expansion board containing a bus master desires to gain control of the system, it activates the slot's master request (`MREQx`) line. Each EISA slot has its own `MREQ` line. The `x` corresponds to the physical location of the slot on the board. The EISA standard allows each adapter card to be addressed independently. The EISA standard makes provisions for up to 15 slots to exist in a system.

Micro Channel (PS/2) Systems

The Micro Channel Architecture (MCA)—invented, trademarked, and patented by IBM—provides computing potential more closely related to mainframe computers. This design has provisions that accommodate multitasking, as well as true, parallel processing. The MCA standard has built-in, bus-sharing provisions that allow up to eight microprocessors, and eight other bus masters to operate within the system.

In developing the Micro Channel Architecture, IBM disregarded any efforts to remain compatible with the ISA standard architecture. Indeed, the MCA standard stands alone in terms of hardware compatibility. The physical expansion connector is incompatible with the edge connectors on ISA adapter cards. In addition, the interface signals called for in the MCA standard are different in definition as well as layout. Therefore, the only source for Personal System/2 (PS/2) peripheral equipment is IBM, or one of its approved vendors. Conversely, IBM has maintained software compatibility between its PS/2 line and the older PC, XT, and AT lines.

Consider the total Micro Channel signal layout. In the MCA standard, the organization of the signal lines can be broken down into three primary sections. First, there is a section of basic 8-bit signals (A/B01–A/B45), located at the rear of the card. This section is followed by a notch in the edge connector that is two contacts wide. The second section of signal lines is the 16-bit extension that runs between contact positions A/B48 and A/B58. The final section of standard signals is the 32-bit extension that runs from positions A/B59 to A/B89.

The MCA bus standard also provides signal and connector specifications for a pair of auxiliary interface connections. These are the auxiliary video extension and the 32-bit matched memory extension. The video extension to the bus is a small, auxiliary connector (usually included with only one of the expansion slots) that allows access to the system's built-in VGA circuitry.

The 32-bit matched-memory extension allows I/O cards with higher-speed capabilities to signal the microprocessor that they can operate faster, which, in turn, causes the system to transfer the information 25% faster.

Many of these signals are similar to those already discussed in association with the PC Bus, the ISA bus, and the EISA bus. Of course, the most notable features of the interface are that the address and data buses have been expanded to the desired 32 bits.

The MCA design includes a single-channel analog audio line that provides audio output, with a frequency range of nearly 10KHz, that approaches the quality of an FM radio signal.

The video extension connector is normally added to one of the slots in the MCA system. This extension allows video enhancement cards to have access to the VGA-compatible circuitry built in to MCA system boards. The extension's signals must be applied to D/A converters before it can be used to drive a VGA monitor.

Another important aspect of the interface layout to notice is that it incorporates a high number of ground lines placed adjacent to high-speed signal lines. This design technique allows the MCA bus to operate at much higher speeds than possible before, while generating much less RFI. The MCA bus is capable of operating at frequencies as high as 80MHz without generating unacceptable levels of RFI.

MCA Bus Operations

Many of the new signal lines in the MCA bus are dedicated to coordinating bus activities between the different microprocessors and bus masters that can be combined in the Micro Channel-based system. These lines arbitrate between the different devices wanting to gain access to the bus to keep them from interfering with one another.

During the configuration process, the system creates a set of disk files, called the adapter description files, in which it stores each expansion card's assigned priority level along with other information related to the card. The individual expansion cards supply this information when their slot's `CARDSETUP` line is activated. Likewise, the expansion card confirms that it is at its correct address by activating its `Card Selected Feedback` line during configuration and diagnostic situations.

When the system boots up, the software routine compares the contents of the disk files with information stored in the system's CMOS setup memory to see that the hardware's configuration has not changed. The bootup software also reads a section of firmware from each expansion card to obtain the ID numbers. These numbers are assigned to the cards by their manufacturers when they are made. The startup software compares the information found on the cards with the CMOS and disk-file configuration information to ensure that the current setup is correct. The card's firmware also contains information concerning its installation and use within the system.

During normal operation of the system, MCA-compatible cards routinely exchange information with system devices concerning their bus sizes. The intelligent devices that control the system signify the size of the information they are placing on the data bus by activating either the `Card Data Size 16` or `Card Data Size 32` line. The cards respond by activating either the `Data Size 16 Return` or

`Data Size 32 Return` line. Similarly, the system's intelligent devices can specify whether they are using 24-bit, 80286-compatible, or extended 32-bit, 80386-compatible addresses by activating (or not activating) the `Memory Address Enable 24` line.

Proprietary Bus Designs

Although other advanced bus specifications added more data lines to their expansion slots, they continued to operate at about 8 to 10MHz. These low bus speeds were required to ensure that compatibility with older I/O cards was maintained. Local buses are designed to improve system performance by allowing the devices attached to the bus to operate directly with the system's microprocessor. These devices operate at speeds nearly equal to that of the microprocessor's clock speed. The most famous local buses are the PCI and VESA buses.

PCI Local Bus

Consider the Peripheral Component Interconnect (PCI) local bus components. The main component in the PCI bus is the PCI bus controller, called the `host bridge`. This device monitors the microprocessor's address bus to determine whether addresses are intended for devices in a PCI slot or one of the board's other types of expansion slots. Each PCI adapter can perform up to eight different functions and multiple bus-mastering cards on the bus.

In PC-compatible systems, the PCI bus normally coexists with the ISA bus. The PCI portion of the bus structure functions as a `mezzanine bus` removed from the system's main bus system. Figure 6.46 depicts a PCI-to-ISA bridge that allows ISA adapters to be used in the PCI system. Other bridge devices can also be used to accommodate either EISA or MCA adapters.

The host bridge routes 32-bit PCI data directly to the expansion slots through the local bus. These transfers occur at speeds compatible with the microprocessor. The host bridge routes non-PCI data to the ISA bridge that converts it into a format compatible with the expansion slot. In the case of ISA slots, the data is converted from 32 bits to the 16-bit ISA format. These transfers occur at typical ISA bus speeds. In EISA and MCA machines, the data is just rerouted in 32-bit format.

The PCI bus specification also uses the host bridge to buffer its bus lines. It supports data buffering for `burst-write operations`. A burst-write operation is performed when the processor supplies a beginning address and then outputs only data to be written to consecutive address locations. The PCI bus will even support burst writes for systems that include a microprocessor, such as an 80386, that doesn't have burst-write capability. Unlike other designs which have a fixed length for a burst write, PCI burst writes can be of indefinite length.

Consider the pin-out of a PCI connector. The PCI local bus specification uses multiplexed address and data lines to conserve the pins of the basic 62-pin PCI connector. Within this connector are signals for control, interrupt, cache support, error reporting, and arbitration.

The PCI bus uses 32-bit address and data buses (AD0–AD31). Its specification, however, also defines 64-bit multiplexed address and data buses for use with 64-bit processors such as the Pentium. The `CLK` line was originally defined for a maximum clock frequency of 33MHz, and a 132MB/second transfer rate, but it can be used with microprocessors operating at higher clock frequencies (66MHz under the PCI 2.1 specification).

The `Request` (REQ) and `Grant` (GNT) lines provide arbitration conventions for bus-mastering operations. The arbitration logic is contained in the host bridge. To allow for faster access, a bus master can request use of the bus while the current bus cycle is in progress. When the current bus cycle ends, the master can immediately begin to transfer data, assuming the request has been granted.

The bus master uses bus commands to specify the type of transaction to be performed. These commands are encoded on the `Command/Byte Enable` lines (C/BE0–C/BE3). These four lines are driven with command information during the first portion of the bus cycle, called the `address phase`. The lines then switch to byte-enable lines for the remaining portion of the bus cycle. The byte-enable pins duplicate the operation of the byte enables on the 80386DX, 80486, and Pentium microprocessors. The commands issued by the bus master include interrupt acknowledge, I/O read, I/O write, memory read, memory write, configuration read, and configuration write.

PCI Configuration

The PCI standard is part of the Plug and Play hardware standard. As such, the system's BIOS and system software must support the PCI standard. Although the PCI function is self-configuring, many of its settings can be viewed and altered through the CMOS Setup utility. Figure 6.38 depicts the PCI PnP configuration information from a typical BIOS.

Figure 6.38

PCI configuration settings.

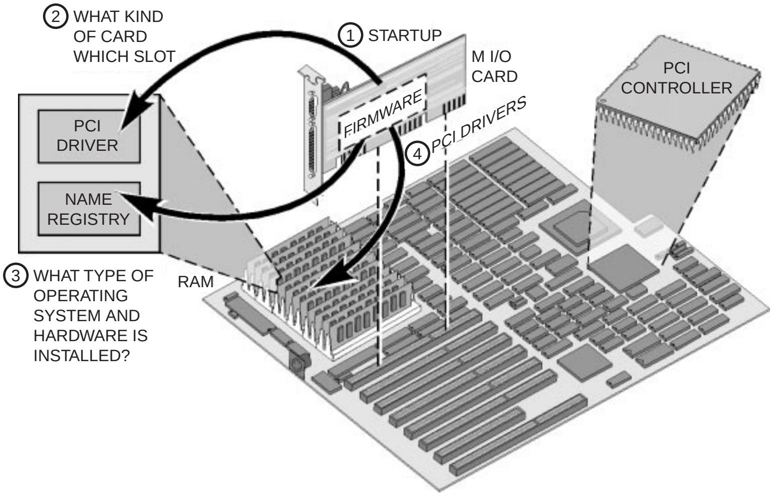
```
                         ROM PCI/ISA BIOS (P155TVP4)
                              PNP AND PCI SETUP
                           AWARD SOFTWARE, INC.

Slot 1 (Right) IRQ   : Auto           DMA 1 Used By ISA       : No/ICU
Slot 2 IRQ           : Auto           DMA 3 Used By ISA       : No/ICU
Slot 3 IRQ           : Auto           DMA 5 Used By ISA       : No/ICU
Slot 4 IRQ           : Auto
PCI Latency Timer    : 32 PCI Clock   ISA MEM Block BASE      : No/ICU

                                      NCR SCSI BIOS           : Auto
IRQ 3 Used By ISA    : No/ICU         USB Function            : Disabled
IRQ 4 Used By ISA    : No/ICU
IRQ 5 Used By ISA    : No/ICU
IRQ 6 Used By ISA    : No/ICU
IRQ 7 Used By ISA    : No/ICU
IRQ 8 Used By ISA    : No/ICU
IRQ 9 Used By ISA    : No/ICU
IRQ 10 Used By ISA   : No/ICU
IRQ 11 Used By ISA   : No/ICU
IRQ 12 Used By ISA   : No/ICU         ESC : Quit           ↑↓←→        : Select Item
IRQ 13 Used By ISA   : No/ICU         F1  : Help           PU/PD/+/-   : Modify
IRQ 14 Used By ISA   : No/ICU         F5  : Old Values     (Shift) F2  : Color
IRQ 15 Used By ISA   : No/ICU         F6  : Load BIOS Defaults
                                      F7  : Load Setup Defaults
```

During bootup, the PCI PnP-compatible BIOS checks the system for devices installed in the expansion slots to see what types they are, how they are configured, and which slots they are in. For PnP-compatible PCI cards, this information is held in a ROM device on the adapter card. As described in the "Bootup" section of Chapter 3, the BIOS reads the information from all the cards and then assigns each adapter a `handle` in the PnP registry. It then stores the configuration information for the various adapters in the registry as well. This process is described in Figure 6.39. Next, the BIOS checks the adapter information against the system's basic configuration for resource conflicts. After evaluating the requirements of the cards and the system's resources, the PnP routine assigns system resources to the cards as required.

Depending on the CMOS settings available with a particular PCI chip set, the startup procedure may be set up to configure and activate all the PnP devices at startup. With other PCI chip sets, it may also be possible to check all cards, but only enable those actually needed for startup. Some CMOS routines may contain several user-definable PCI configuration settings. Typically, these settings should be left in default positions. The rare occasion for changing a PCI setting occurs when directed to do so by a product's Installation Guide.

Figure 6.39

PCI information acquisition.



Systems may theoretically contain an unlimited number of PCI slots. However, a maximum of four slots are normally used on a system board due to signal loading considerations. The PCI bus includes four internal interrupt lines (INTa through INTd, or INT1 through INT4) that allow each PCI slot to activate up to four different interrupts. PCI interrupts should not be confused with the system's IRQ channels, although they can be associated with them if required by a particular device. In these cases, IRQ9 and IRQ10 are typically used.

The `Latency Timer` value in the figure refers to the length of time a PCI device can maintain control of the bus after another device has requested it. This feature is necessary because the PCI bus runs much faster than the ISA bus. The latency timer setting defines how long the PCI bus will wait for the ISA bus. Values assigned to the timer range from 0 to 255, with the lower number representing quicker access to the bus.

In the sample CMOS configuration screen, a block of memory in the UMA can be set aside as an `ISA memory block base` area. This feature allows the system to temporarily write data to a buffer memory so that the microprocessor is not disturbed. Data from the buffer can be handled as system priorities allow.

VESA Local Bus

Figure 6.40 illustrates the flow of information through the VL bus-based computer. It also indicates data transfer priority levels. Like the PCI Bus, the VL-bus controller monitors the microprocessor's control signals and addresses on the local bus to determine what type of operation is being performed and where the address is located in the system.

The highest level of activity occurs between the microprocessor and the system's cache memory unit. The second level of priority exists between the microprocessor and the system's DRAM memory unit. The third-priority level is between the microprocessor and the VL-bus controller. The final-priority level exists between the VL-bus controller and the non-VESA bus controller.

VL-bus data is passed to the VESA slots on the local bus in 32-bit format at VL-bus speeds. The VL-bus controller passes non-VESA data to the ISA bus controller to be applied to the ISA expansion slots. These transfers are carried out in 16-bit ISA format, at ISA-compatible speeds.

The VL bus also defines the operation of devices connected to the bus, and classifies them as either `local bus controller`, `local bus master`, or `local bus target`. The local bus controller arbitrates requests for use of the bus between the microprocessor and local bus masters. A local bus master is any device, such as a SCSI controller, that is capable of initiating data transfers on the VL bus. A local bus target is any device capable of answering requests only for a data transfer. The data transfer can be either a read or a write operation. The VL bus also supports `burst writes` for faster transfers and increased efficiency.

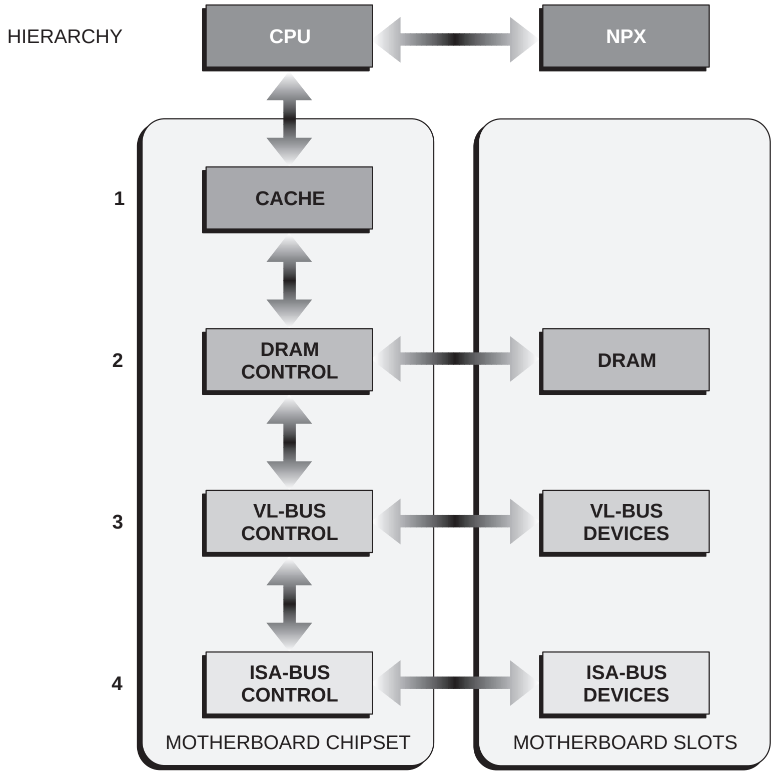


Figure 6.40

VL bus block diagram.

The VL-bus connection pin-out is shown in Figure 6.41.

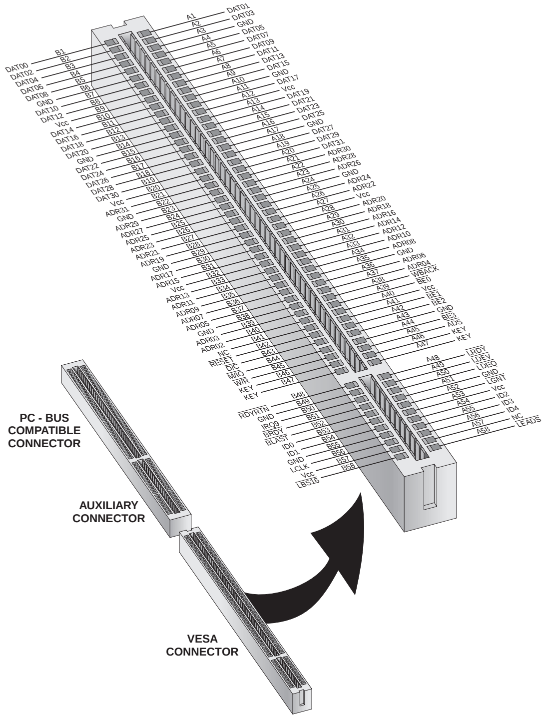
The VL local bus clock can operate at speeds up to 66MHz. If the VL-Bus devices are installed into an expansion slot, however, the maximum frequency allowed is 50MHz. The 66MHz clock rate can be used only if the VL-bus device is built directly on the system board.

Although the PCI bus is more widely used on Pentium system boards, the VL bus does have the advantage that it offers higher performance, and lower costs, than similar boards with PCI buses.

In addition, the VL bus is typically implemented in such a way that 16-bit ISA cards can use the traditional part of the expansion slot.

Figure 6.41

VL bus pin-out



PCMCIA Slots

As more and more desktop users began to use laptop and notebook computers, they demanded that additional peripheral systems be included. With the limited space associated with these units, it became clear that a new method for installing options would need to be developed. At first, laptop and notebook manufacturers included proprietary expansion connections for adding such devices as fax/modems, additional memory, and additional storage devices.

In 1989, the PCMCIA bus was introduced with the 68-pin JEIDA connector standard. A small form-factor, expansion-card format was also adopted for use. This format was derived from earlier laptop/notebook memory-card designs. The definitions of the connector's 68 pins, listed in Table 6.10, allow the interface to be used for a wide variety of peripheral devices.

Table 6-10

68-pin connector definitions.

Pin	Name	Description	Pin	Name	Description
1	GND	Ground	35	GND	Ground
2	D3	Data Bit 3	36	CD1	Card Detect 1
3	D4	Data Bit 4	37	D11	Data Bit 11
4	D5	Data Bit 5	38	D12	Data Bit 12
5	D6	Data Bit 6	39	D13	Data Bit 13
6	D7	Data Bit 7	40	D14	Data Bit 14
7	CE1	Card Enable 1	41	D15	Data Bit 15
8	A10	Address Bit 10	42	CE2	Card Enable 2
9	OE	Output Enable	43	RFSH	Refresh Input
10	A11	Address Bit 11	44	IORD	I/O Read Strobe
11	A9	Address Bit 9	45	IOWR	I/O Write Strobe
12	A8	Address Bit 8	46	A17	Address Bit 17
13	A13	Address Bit 13	47	A18	Address Bit 18
14	A14	Address Bit 14	48	A19	Address Bit 19
15	WE/-PGM	Write Enable	49	A20	Address Bit 20
16	IREQ	Interrupt Request	50	A21	Address Bit 21
17	VCC	Card Power	51	VCC	Card Power
18	VPP1	Programming Supply Voltage 1	52	VPP2	Programming Supply Voltage 2
19	A16	Address Bit 16	53	A22	Address Bit 22
20	A15	Address Bit 15	54	A23	Address Bit 23
21	A12	Address Bit 12	55	A24	Address Bit 24
22	A7	Address Bit 7	56	A25	Address Bit 25
23	A6	Address Bit 6	57	RFU	Reserved
24	A5	Address Bit 5	58	RESET	Card Reset
25	A4	Address Bit 4	59	WAIT	Extend Bus Cycle

continues

Table 6-10 Continued

Pin	Name	Description	Pin	Name	Description
26	A3	Address Bit 3	60	INPACK	Input Port Acknowledge
27	A2	Address Bit 2	61	REG	Register and IO select enable
28	A1	Address Bit 1	62	SPKR	Digital Audio Waveform
29	A0	Address Bit 0	63	STSCNG	Card Status Changed
30	D0	Data Bit 0	64	D8	Data Bit 8
31	D1	Data Bit 1	65	D9	Data Bit 9
32	D2	Data Bit 2	66	10	Data Bit 10
33	IOIS16	IO Port is 16 bits	67	CD2	Card Detect 2
34	GND	Ground	68	GND	Ground

The interface is designed so that it can be inserted into the unit while it is turned on (`hot insertion`). Although the PCMCIA connection scheme was never intended to be used with a full-sized unit, its design is compatible with all the other bus types. As a matter of fact, PC-Card adapters are available for use in desktop and tower units. These slots are often designed so that they can be mounted in a standard disk drive bay of the desktop's case.

The standard defines a methodology for software programmers to write standard drivers for PC-Card devices. The standard is referred to as `Socket Services` and provides for a software head to identify the type of card being used, its capabilities, and its requirements. Its software can be executed directly on the card (instead of moving it into RAM for execution). This is referred to as `execute in place` mode. In addition, its cards can use the same file allocation system used by floppy and hard disk drives. This also increases the ease with which programmers can write code for PCMCIA devices.

System Board Upgrading

There are typically five serviceable components on the system board. These include:

- the microprocessor
- the RAM modules
- the CMOS Backup battery
- the ROM BIOS IC(s)
- the cache memory

Of the five items listed, three - the microprocessor, the RAM modules and the cache memory - can be exchanged to increase the performance of the system. These devices are normally mounted in sockets to make replacing or upgrading them an easy task.

Great care should be taken when exchanging these parts to avoid damage to the ICs from *Electrostatic Discharge* (ESD). ESD prevention is covered in detail in Chapter 13 - Preventive Maintenance. In addition, care should be taken during the extraction and replacement of the ICs to avoid mis-alignment and bent pins. Make sure to correctly align the IC's pin #1 with the socket's pin #1 position. In the case of microprocessors that plug into standard socket types (also referred to as *Low Insertion Force* - or *LIF* sockets), the force required to insert them may over-stress the system board if not properly supported.

As stated earlier in this chapter, microprocessor manufacturers have devised upgrade versions for virtually every type of microprocessor in the market. It is also common for clone microprocessors to be pin compatible with older Intel socket designs. This strategy allows the end user to not only realize a speed increase by upgrading, but a processing power increase as well.

Upgrading the processor is a fairly easy operation after gaining access to the system board. Simply remove the microprocessor from its socket and replace it with the upgrade. Two items must be observed when changing the microprocessor:

- make sure the replacement is pin out compatible with the original

- make sure to properly orient the new processor in the socket so that its pin #1 matches the socket's pin #1

Upgrading system board memory is also a fairly simple process. Having more RAM on board allows the system to access more data from extended or expanded memory, without having to access the disk drive. This of course, speeds up system operation considerably. Normally, upgrading memory simply amounts to installing new memory modules in vacant SIMM or DIMM slots. If the slots are already populated, it will be necessary to remove them to install faster, or higher capacity modules.

The system board User's Guide should be consulted to determine what speed the memory devices must be rated for. You should be aware that RAM and other memory devices are rated in access time instead of clock speed. Therefore, a -70 nanosecond (ns) RAM device is faster than an -80 nanosecond device. The guide should also be checked for any memory configuration settings that must be made to accept the new memory capacity.

If the system has socketed cache memory, some additional performance can be gained by optimizing the cache. Upgrading the cache on these system boards normally only requires that additional cache ICs be installed in vacant sockets. If the sockets are full but the system's cache size is less than maximum, it will be necessary to remove the existing cache chips and replace them with faster, higher-capacity devices. Make sure to observe the pin #1 alignment as well as check the system board's User's Guide for any configuration jumper changes.

System Board Troubleshooting

Objective

Troubleshooting problems related to the system board can be difficult to solve due to the system board's relative complexity. So many system functions at least partially rely on the system board that certain symptoms can be masked by other symptoms.

As with any troubleshooting procedure, begin by observing the symptoms produced by bootup and operation. Observe the steps that lead to the failure and determine under what conditions the system failed. Were there any unusual operations in progress? Note any error messages or beep codes.

Objective

Try any obvious steps, such as adjusting brightness controls on a dim monitor or checking for loose connections on peripheral equipment. Check power switch settings on every system component. Retry the system several times to observe the symptoms clearly. Take the time to document the problem—write it down.

Refer to the system board and peripheral units' User's Guides to look for configuration problems. Check the CMOS Setup utility for configuration problems. Diagnose the problem to a section of the system (in this case, the system board). In Pentium systems, check the Advanced CMOS Setup parameters to make certain that all of the appropriate system board enabling settings have been made.

If possible, run a software diagnostics package to narrow the possible problem causes. Remember that the microprocessor, RAM modules, ROM BIOS, CMOS battery, and possibly cache ICs are typically FRU units on the system board. If the diagnostics program indicates a number of possible bad components, replace them one at a time until the bad unit has been isolated. Then insert any possible good units back into the system and check them. The possibility of bad software should also be considered when multiple `Field Replaceable Unit` (FRU) problems are indicated.

If possible, back up the contents of the hard drive before removing the system board. Record the CMOS configuration settings, along with the settings of all jumpers and switches, before exchanging the system board.

System Board Symptoms

So much of the system's operation is based on the system board that it can have several different types of symptoms. Typical symptoms associated with system board hardware failures include the following:

- The ON/OFF indicator lights are visible. The display is visible on the monitor screen, but there is no disk drive action and no bootup.
- The ON/OFF indicator lights are visible, the hard drive opens up, but the system appears dead and there is no bootup.

- System locks up during normal operation.
- A beep code with 1, 2, 3, 5, 7, or 9 beeps is produced by the system.
- A beep code of 1 long and 3 short beeps is produced by the system.
- System will not hold date and time.
- An `8042 Gate A20 Error` message, indicating an error getting into protected mode.
- An `Invalid Switch Memory Failure` message.
- A `DMA Error` message, indicating that the DMA controller failed page register test.
- A `CMOS Battery Low` message, indicating the failure of CMOS battery or CMOS checksum test.
- A `CMOS System Option Not Set` message, indicating the failure of CMOS battery or CMOS checksum test.
- A `CMOS Checksum Failure` message, indicating CMOS battery low or CMOS checksum test failure.
- A 201 error code is displayed onscreen, indicating a RAM failure.
- A `Parity Check` error message onscreen, indicating a RAM error.

Typical symptoms associated with system board setup failures include the following:

- A `CMOS Inoperational` message, indicating failure of CMOS shutdown register.
- A `Display Switch Setting Not Proper` message, indicating failure to verify display type.
- A `CMOS Display Mismatch` message, indicating failure of display type verification.
- A `CMOS Memory Size Mismatch` message, indicating system configuration and setup failure.
- A `CMOS Time & Date Not Set` message, indicating system configuration and setup failure.

- An IBM-compatible error code displayed onscreen, indicating that a configuration problem has occurred.

Typical symptoms associated with system board I/O failures include the following:

- Speaker doesn't work.
- Keyboard does not function after being replaced with known good unit.

Most of the hardware problems that occur with computers, outside of those already described, involve the system board. Because the system board is the center of virtually all the computer's operations, it's only natural that it will need to be checked at some point in most troubleshooting efforts. The system board normally marks the end of any of the various troubleshooting schemes given for different system components. It occupies this position for two reasons. The first reason is that the system board supports most of the other system components, either directly or indirectly. Second, it is the system component that requires the most effort to replace and test.

Other System Board Problems

In addition to containing the circuitry that directs all the system's operations, the system board contains a number of other circuits that the rest of the system's components depend on. These include the system's DRAM memory (which all software programs use) and the system's data, address, and signal buses. The part of the buses you are most familiar with are the expansion slots.

Problems with key system board components will produce symptoms similar to those described for a bad power supply. As a matter of fact, part of the Power Supply Problem Isolation procedure (Procedure 18 in the Hands-on Lab Procedures) refers to checking the system board. Both the microprocessor and the ROM BIOS can be sources of these problems. Both should be checked by substitution when dead system symptoms are encountered but the power supply is good.

In the case of the microprocessor, the system may issue a slow, single beep, with no display or other I/O operation. This indicates that an internal error has disabled a portion of the processor's internal circuitry (usually the internal cache). Internal problems may also allow the microprocessor to being processing, but fail as it attempts operations. Such a problem results in the system continuously count RAM during the bootup. It may also lock up while counting RAM. In either case, the only method of repair is to replace the microprocessor.

The microprocessor should also be checked if its fan has not been running, but the power is on. This may indicate that the microprocessor has been without adequate ventilation and overheated. When this happens, it will be necessary to replace the fan unit and the microprocessor. Check to make certain that the fan works correctly or a second microprocessor will be damaged.

Like the microprocessor, a bad or damaged ROM BIOS will typically stop the system dead. When a dead system board is encountered, examine the BIOS chip (or chips) for physical damage. If these devices over heat, it is typical for them to crack, or blow a large piece out of the top of the IC package. Another symptom of a damaged BIOS, is indicated by the bootup moving into the CMOS configuration, but never returning to the bootup sequence. In any case, the defective BIOS must be replaced with a version that matches the chip set used by the system.

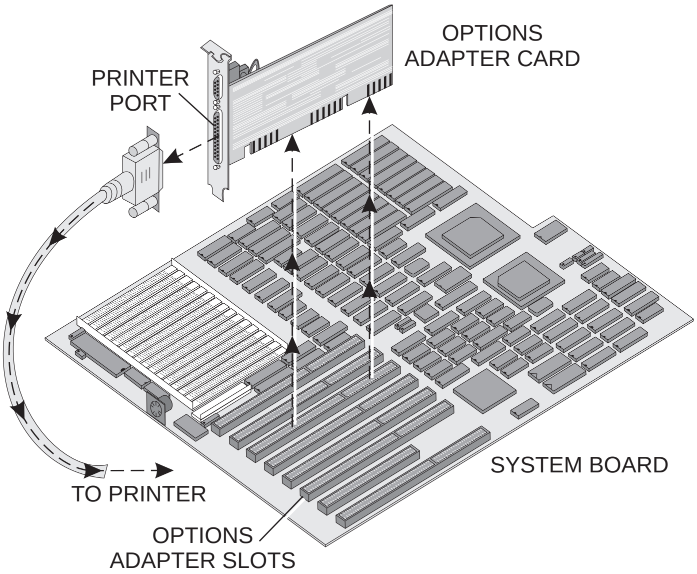
The system board's memory is a very serviceable part of the system. Many of the symptoms listed in the previous paragraphs pertain to the system's RAM. A class of memory errors, called `soft memory errors`, are caused by infrequent and random glitches in the operation of applications and the system. These events can be cleared just by restarting the system. However, the other errors listed will normally relate to `hard memory errors`, and require that memory units be checked by substitution, or that a configuration jumper be reset.

Because all of the system's options adapters connect to the buses through the expansion slots, failure of any component attached to one of the slots can prevent information movement between other components along the bus. In this case, it will be necessary to remove the offending component from the bus before any operation can proceed.

A number of other optional devices may be added to the system just by installing an appropriate options adapter card in one of the system board's expansion slots, and then connecting the option to it. Figure 6.42 illustrates the flow of information between the system board and a typical connection port (a parallel printer port) located on an options adapter card.

Figure 6.42

Moving information to an I/O port.



An often overlooked output device is the system's speaker. Unlike other I/O devices, all the circuitry that controls the speaker is contained on the system board. Therefore, there are only a few reasons that the speaker should fail. These reasons include the speaker itself is defective, the speaker circuitry on the system board is defective, the speaker is unplugged from the system board, or the software is failing to drive the speaker circuits. The speaker-related components are depicted in Figure 6.43.

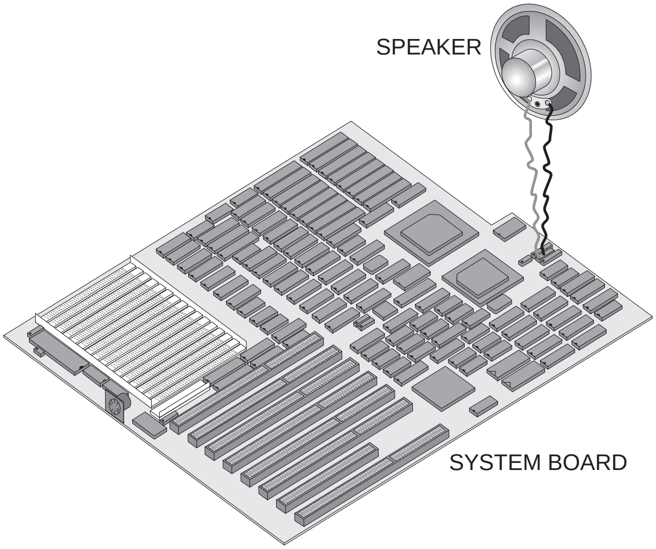
The keyboard is another I/O device supported directly from the system board. When examining keyboard problems, there are only three items to check: the keyboard, the system board, and the keyboard driver software.

Configuration Checks

Observe the bootup RAM count to see that it is correct for the amount of physical RAM actually installed in the system. If not, swap RAM devices around to see whether the count changes. Use logical movement of the RAM devices to locate the defective part.

Figure 6.43

Speaker-related components.



Normally, the only time a configuration problem occurs is when the system is being set up for the first time, or when a new option is installed. The other condition that will cause a configuration problem involves the system board's CMOS backup battery. If the battery fails, or has been changed, the contents of the CMOS setup will be lost. After replacing the battery, it is always necessary to run the CMOS Setup utility to reconfigure the system.

The values stored in this memory must accurately reflect the configuration of the system; otherwise, an `error` will occur. These values can be accessed for change by pressing the Ctrl and Del keys (or some other key combination) simultaneously, during the bootup procedure.

Typically, if the bootup process reaches the point where the system's CMOS configuration information is displayed onscreen, it can be assumed that no hardware configuration conflicts exist in the system's basic components. After this point in the bootup process, the system begins loading drivers for optional devices and additional memory. If the error occurs after the CMOS screen is displayed and before the bootup tone, it will be necessary to clean boot the system and single step through the remainder of the bootup sequence.

Software Checks

Boot up the system and start the selected diagnostic program, if possible. Try to use a diagnostic program that deals with the system board components. It should include memory, microprocessor, interrupt, and DMA tests.

Run the program's `System Board Tests` function and perform the equivalent of the ALL tests function. These types of tests are particularly good for detecting memory errors, as well as interrupt and DMA conflicts. Note all the errors indicated by the tests. If a single type of error is indicated, it may be possible to take some corrective actions, such as replacing a memory module or reconfiguring interrupt/DMA settings, without replacing the system board. If more complex system board problems are indicated, however, exit the diagnostic program and use the following Hardware Checks and Installation/Removal procedure to troubleshoot and replace the system board.

The DOS `MEM` command can be used to view the system's memory utilization scheme. It displays both the programs currently loaded into memory, and the system's free memory areas. The `/C` switch can be used with the `MEM` command as a valuable tool to sort out TSR conflicts in upper memory.

The MemMaker utility is used to modify the system's CONFIG.SYS and AUTOEXEC.BAT files, so that device drivers and memory resident programs take up less conventional memory space. This is accomplished by loading these types of programs into the upper-memory area. It requires a microprocessor capable of running in virtual memory mode. MemMaker is often used to resolve memory conflicts in the lower 64KB of memory.

Hardware Removal and Installation

After a problem has been isolated to the system board hardware, it will be necessary to remove and replace it. The removal procedure can be defined in five steps:

1. Remove all external I/O systems
2. Remove the system unit's outer cover
3. Remove the option adapter cards

4. Remove the cables from the system board
5. Remove the system board

During this operation, it is necessary to disconnect several connectors from the old system board and reconnect them to the new system board. The easiest method of handling this is to use tape (preferably masking tape) to mark the wires and their connection points (on the new system board) before removing any wires from the old system board.

Removing All External I/O Systems

Unplug all power cords from the commercial outlet. Remove all peripherals from the system unit. Disconnect the mouse, keyboard, and monitor signal cable from the rear of the unit. Finally, disconnect the monitor power cable from the system unit (or the outlet).

Removing the System Unit's Outer Cover

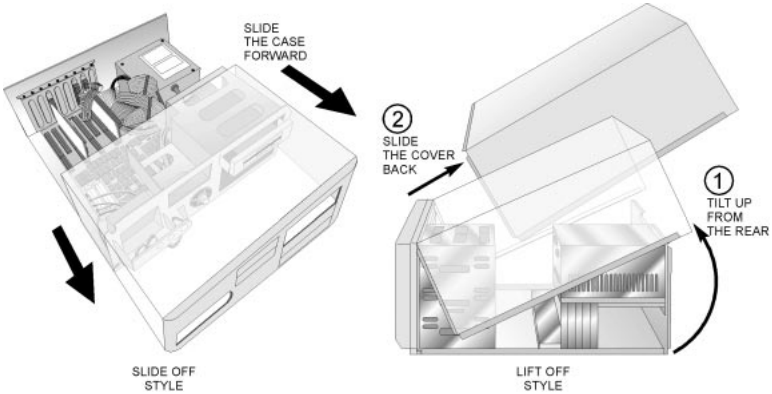
Unplug the 120-Vac power cord from the system unit. Determine which type of case you are working on. If the case is a desktop model, does the cover slide off the chassis in a forward direction, bringing the front panel with it, or does it raise off of the chassis from the rear? If the back lip of the outer cover folds over the edge of the unit's back panel, the lid raises up from the back, after the retaining screws are removed. If the retaining screws go through the back panel without passing through the lip, the outer cover will slide forward after the retaining screws have been removed.

Check to see how many screws hold the outer cover to the chassis. Do not confuse the power supply retaining screws with those holding the back panel. The power supply unit requires four screws. Check for screws along the lower edges of the outer cover that hold it to the sides of the chassis. Remove the screws that hold the cover to the chassis. Store the screws properly.

Remove the system unit's outer cover, as illustrated in Figure 6.44, and set it aside. Slide the case forward. Tilt the case upward from the front and remove it from the unit. Or, lift the back edge of the outer cover to approximately 45 degrees, and then slide it toward the rear of the chassis.

Figure 6.44

Removing the case.



Removing the Option Adapter Cards

A wide variety of peripheral devices are used with PC-compatible systems. These devices communicate with the main system through options adapter cards that fit into slot connectors on the system board. The adapter cards that come in most pre-Pentium PC systems are the video controller card and the MI/O card. The MI/O card contains the disk drive controller circuitry, as well as the system's serial and parallel port connections. The hard- and floppy-disk drives communicate with this card through ribbon cables. These cables have indicator stripes that must be aligned properly on both ends. In Pentium systems, the MI/O card is integrated directly into the system board. It is a good practice to place adapter cards back into the same slots they were removed from, if possible.

Remove the retaining screws that secure the options adapter cards to the system unit's back panel. Remove the video adapter card from the expansion slot. Store the screws properly. Refer to Figure 6.45 to perform this procedure.

If the system uses an MI/O card, disconnect the floppy drive signal cable (smaller signal cable) and the hard drive signal cable (larger signal cable) from the MI/O card. Also disconnect any I/O port connections from the MI/O card. Remove the MI/O adapter card from the expansion slot.

Removing the Cables from the System Board

The system board provides an operator interface through a set of front-panel indicator lights and switches. These indicators and switches connect to the system board by BERG connectors, as depicted in Figure 6.46.

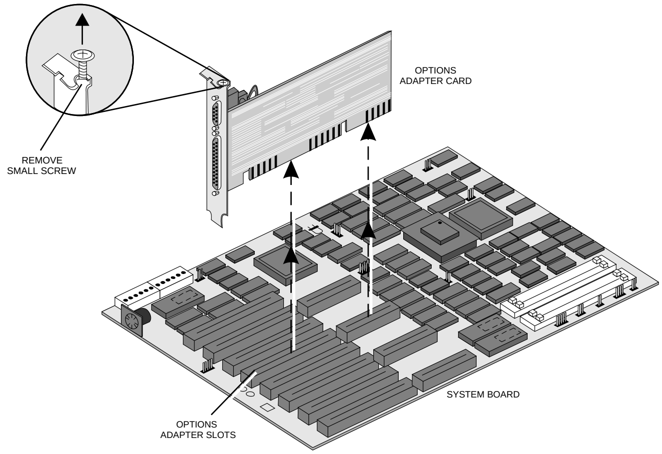


Figure 6.45

Removing adapter cards.

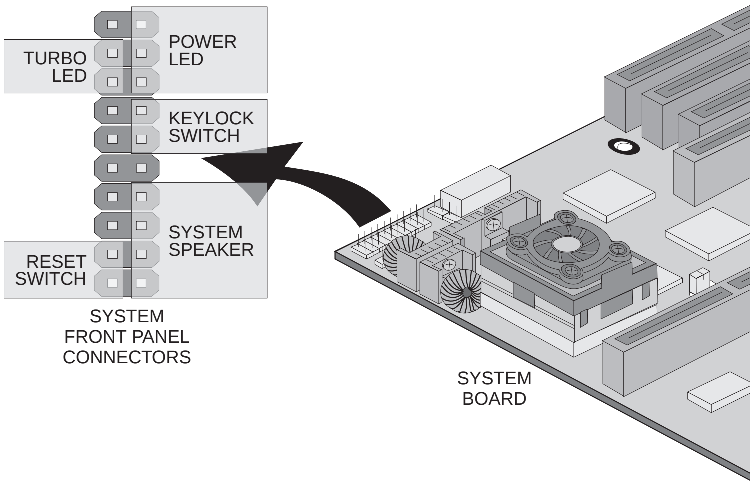


Figure 6.46

Front-panel connections.

The front-panel connectors need to be removed to exchange the system board for a new one. Because it is quite easy to get these connections reversed, make sure that you mark them for identification purposes before removing them from their connection points. Record the color and function of each connection. Trace each wire back to its front-panel connection to determine what its purpose is. This will ensure that they are reinstalled correctly after the exchange is completed.

Disconnect the P8 and P9 power supply connections from the system board as well.

Removing the System Board

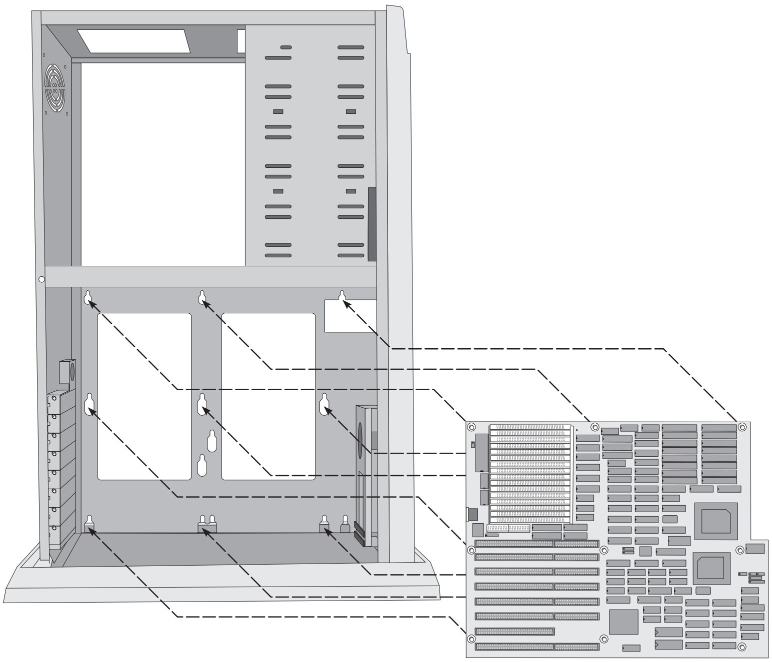
Check the positions of all jumper and switch settings on the old system board. Record these settings and verify their meanings before removing the board from the system. This may require the use of the system board's User's Manual, if available. Remove the retaining screw(s) that secures the system board to the floor of the chassis. Store the screw(s) properly.

In a desktop unit, slide the system board toward the left side of the system unit (as you face the front of the unit) to free its plastic feet from the slots in the floor of the system unit. Tilt the left edge of the board up, and then lift it straight up and out of the system unit.

In a tower unit, slide the system board toward the bottom of the system unit to free its plastic feet from the slots in the side panel. Tilt the bottom edge of the board away from the unit and pull it straight out of the chassis, as shown in Figure 6.47.

Figure 6.47

Removing the system board from a tower case.



Reassembling the System

The system board installation procedure is basically the removal procedure in reverse. However, the CMOS setup and hardware configuration will have to be reset after the new system board has been installed. Also, try to replace the options adapter cards in the same expansion slots they were removed from.

Hardware Checks

If the system's CMOS configuration setup appears to be correct and a system board hardware problem is suspected, the first task is to remove all the externally connected devices from the rear of the system, except for the monitor and keyboard. Try to boot up the system with just the basic I/O options installed.

If the system operates correctly with the external options removed, it is safe to assume that one of them is the cause of the problem. To verify which external device is causing the problem, reconnect the devices, one at a time, until the problem reappears. The last device reinstalled before the problem reappeared is defective. Replace this item and continue reinstalling options, one at a time, until all the options have been reinstalled. If another failure occurs while reinstalling options, replace that option as well. Repair or replace the defective option and return the system to full service.

If none of the external options seem to be causing the problem, the next easiest items to test are the operation and configuration of the hard disk drive. This is accomplished by trying to boot the system to a clean, self-booting disk in the A: floppy drive. This test can be accomplished without removing the outer cover of the system unit.

Check the CMOS setup, if possible, to make sure that the system will check the A: drive as part of the boot sequence. Place a bootable disk (or clean boot disk) in the A: floppy drive and turn the system on. If the system boots to the floppy, there is a failure in the hard drive or controller. Refer to lab procedures 27 and 28 in the Lab Guide to service the HDD section of the system.

Note

> The process for isolating system board problems is reinforced and expanded in the accompanying Hands-on Lab Book in procedure 13.

Checking Inside the System Unit

Remove the system unit's outer cover and then remove all the internal options adapter cards except for the video and disk-drive controller cards from the system board's expansion slots. Try to reboot the system. If the system works correctly with the additional adapter cards removed, it is reasonable to assume that one of them was causing the problem. To verify which option is causing the problem, reinstall them, one at a time, until the problem reappears. The last adapter reinstalled before the problem reappeared is defective. Replace this adapter with a working unit and continue reinstalling adapters, one at a time, until all the cards have been reinstalled. If another failure occurs while reinstalling adapters, replace that unit as well.

If the system boots up with the internal options removed, it is reasonable to assume that one of them is the cause of the problem. Therefore, you should reinstall the internal options one at a time until the problem reappears. As always, the last option reinstalled before the problem returned is defective. Replace the defective device or card with a new one, reinstall any options removed from the system, and return the system to full service.

Checking Basic Components

If the system still refuses to boot up, the basic adapter cards should be checked next. It doesn't really matter which card is checked first unless some symptom points to a particular card. In older units, try the MI/O card first, because it contains a variety of system functions. Turn the system off, and remove the disk-drive controller card from the system board's expansion slot. If the system produces an HDD or FDD error message on the screen, exchange the disk drive controller (in pre-Pentium systems) with a known good one. Make certain to mark the cable and its connection point to ensure proper reconnection after the exchange. Try to reboot the system with the new disk drive controller installed.

If problems continue, remove the video controller card from the system board's expansion slots and turn the system on. Does the FDD activity light come on? If so, exchange the video controller card with a known good one of the same type.

Check for +5 and +12 Vdc on the system board, as illustrated in Figure 6.48. If these voltages are missing, turn the system off, disconnect power to all disk drives, and swap the power supply unit with a known good one.

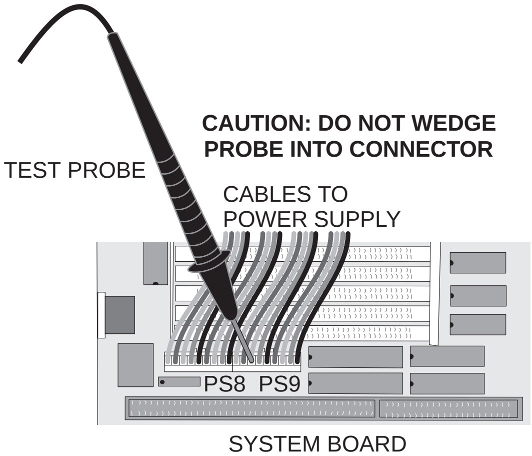


Figure 6.48

The system board voltage check location.

Exchanging the System Board

If the system still won't boot up, remove the video and disk-drive controller cards from the system board's expansion slots. Disconnect the system board from the power supply unit (P8–P9) and the system board/front-panel connections. Take care to mark any connection removed from the system board, and its connection point, to ensure proper reconnection. Exchange the system board with a known good one. Reconnect all the power-supply and front-panel connections to the system board. Reinstall the video and disk-drive controller cards in the expansion slots and try to reboot the system.

Reconfigure the system board to operate with the installed peripherals. Reseat the video and disk-drive controller cards in the system unit. Reset the CMOS Setup utility to match the installed peripherals, and turn the system on.

When the system boots up, reinstall any options removed from the system and replace the system unit's outer cover. Return the system to full service and service the defective system board. If the system still does not boot up, retest all the system components one at a time until a cause is found. Check the small things such as cable connections and key switches carefully.

There are a few serviceable items on the system board. These include the RAM modules, the microprocessor (and its cooling fan), the ROM BIOS chip(s), and the system battery.

The RAM modules can be swapped out in a one-at-a-time manner, to isolate defective modules. These modules are also swapped out when a system upgrade is being performed. The burn-in tests in most diagnostic packages can be helpful in locating borderline RAM modules.

The microprocessor can be exchanged easily on most system boards. Only the 80386SX is a soldered-in device, so it would be much more difficult to exchange and would not likely be worth the expense involved. However, the fact that most microprocessors, as well as the BIOS chips, are mounted in sockets brings up another point. These items should be pulled, and reseated in their sockets if they seem to be a possible cause of problems. Sockets are convenient for repair and upgrade purposes, but they can also attract corrosion between the pins of the device, and those of the socket. After some time, the electrical connection may be too poor for the device to operate properly.

Corrosion can also affect the system clock over time. If a system refuses to maintain time and date information after the backup battery has been replaced, check the contacts of the holder for corrosion. There are two types of batteries commonly used for CMOS backup. These are nickel-cadmium (NiCad) and Lithium batteries. Of the two, NiCads have historically been the more favored. Conversely, lithium batteries are gaining respect due to their long life capabilities when installed in systems designed to recharge lithium batteries. However, lithium battery life is noticeably short when they are installed in systems designed for the higher current drain NiCads. Therefore, the correct type of battery should always be used to replace a system board battery.

Summary

The system board is the main component of any personal computer system. This chapter has examined the major components that make up typical PC-compatible system boards. These items include microprocessors, memory types, microprocessor support systems, and expansion buses.

All the microprocessors associated with PC-compatibles have followed in the basic structures established by the 8088 in the original IBM PC. In like manner, the system's support devices have been integrated into smaller and smaller packages, but have not changed significantly in their operation since the PC AT. The most significant changes in PCs have occurred in the area of expansion slots. Larger and faster expansion slots are always in demand because they allow faster, more-powerful peripheral devices to be used with the system. This is important because the transmission of data back and forth between peripherals and the system is normally the biggest bottleneck in the system's operation.

The beginning of this chapter discussed the theory related to the system board's components, and the final section of the chapter dealt with troubleshooting and replacing defective system boards or their FRU components.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each item and can perform the tasks listed there. Afterward, answer the following review questions to verify your knowledge of the information.

Lab Exercise

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 12—Memory Optimization, 13—System Board Problem Isolation, and 19—Speed Indicators.

Review Questions

1. What categories of devices are checked before actually checking the system board? Why is this?
2. Describe three different types of symptoms associated with system board failures.
3. Why should you try to boot the system from a clean, self-booting disk before removing the system unit's cover?
4. From the troubleshooting information in this chapter, what is indicated if the bootup reaches the CMOS setup screen before failing? What does this information have to do with troubleshooting system board problems?
5. In the hardware troubleshooting checks, is there any reason that an MI/O card is tested before the video card?
6. Where is the CMOS setup stored in the system? What does it hold?
7. How many hardware interrupt channels are there in a PC- or XT-compatible system?
8. What maximum clock frequency should be applied to an 80286-12 microprocessor?
9. Where can interrupt request #1 be found in a PC compatible system?
10. How many total addresses can the 8088 microprocessor select when executing memory operations?
11. How many hardware interrupt channels are available in an AT-compatible system?
12. What function does IRQ-0 play in a PC-compatible system?
13. How many DMA channels are included in a PC- or XT-compatible system?

14. What function does DMA channel 2 serve in a PC-compatible system?
15. Describe the operation of the system when it receives a DMA request.
16. What function does IRQ 2 serve in an AT-compatible system?
17. Which IRQ channel services the FDD in PC-compatible systems?
18. Name two events that can mark the end of a DMA transfer.
19. What items can cause an NMI to occur?
20. List the functions of the system's timer/counter outputs.
21. Name three advantages of the 80286 microprocessor over the 8088.
22. List at least four microprocessor support circuits common to all PC-compatibles.
23. List the three major sections of the 80486 microprocessor.
24. What is the maximum allowable clock frequency that should be used to drive an 80486-33 microprocessor?
25. Name two advantages of using chip sets to design circuit boards.

Review Answers

1. External and internal options should be checked before removing the system board. The system board is typically the most difficult and expensive part of the system to replace. For more information, see the section titled "Hardware Checks."
2. Types of symptoms associated with system board failures include hardware, setup, and system board I/O failures. For more information, see the section titled "System Board Symptoms."
3. The problem may actually be located in the HDD configuration. This condition can be cured without removing the outer cover. For more information, see the section titled "Hardware Checks."
4. If the bootup process reaches the system's CMOS configuration information display, it can be assumed that no hardware configuration conflicts exist in the system's basic components. For more information, see the section titled "Configuration Checks."

5. No. The order in which the required internal options adapters is checked does not matter. For more information, see the section titled “Checking Basic Components.”
6. In the registers of the RTC. This section of circuitry is normally stored in the integrated peripheral controller on the system board. For more information, see the section titled “Real-Time Clock.”
7. Eight. For more information, see the section titled “System Board Evolution.”
8. The maximum external clock frequency for the 286-12 is 24 MHz. For more information, see the section titled “The 80286 Microprocessor.”
9. At the system board’s keyboard controller chip. This IRQ channel is activated when the keyboard buffer becomes full. For more information, see the section titled “Interrupt Controllers.”
10. A total of 1,048,576 bytes (1MB) of memory addresses. For more information, see the section titled “8088 Pin Assignments.”
11. Fifteen. (One of the 16 channels is used to cascade the first controller to the second.) For more information, see the section titled “Interrupt Controllers.”
12. It drives the system’s time-of-day clock. For more information, the section titled section “System Board Evolution.”
13. Four. For more information, see the section titled “System Board Evolution.”
14. It provides the system’s FDD DMA channel. For more information, see the section titled “System Board Evolution.”
15. When the DMA controller receives a DMA request, it issues a hold request to the system’s microprocessor. The microprocessor responds by finishing its current instruction and placing its bus pins in a high-impedance state (effectively disconnecting them from the bus). It follows this by notifying the DMA controller that it can use the buses to conduct high-speed data transfers. For more information, see the section titled “A Typical DMA Operation.”

16. It is unused in PC- and PC/XT-compatible systems. In AT-compatible systems, it is the cascaded IRQ input for the secondary PIC device. For more information, see the section titled "System Board Evolution."
17. IRQ-6 is dedicated to the FDD. For more information, see the section titled "System Board Evolution."
18. The terminal count (T/C) signal from the DMA controller or an end-of-process (EOP) count from the FDC controller can be generated. If the transfer is correct, both signals will be generated at the same time. For more information, see the section titled "A Typical DMA Operation."
19. I/O channel check and parity check errors can generate NMI failures. For more information, see the section titled "Non-Maskable Interrupts (NMI)."
20. System time-of-day clock, DRAM refresh generator, and tone generator for the system speaker. For more information, see the section titled section "Timer/Counter Channels."
21. Its data bus is 16 bits wide (as opposed to 8 in the 8088), it supports `multiuser` and `multitasking` operations, and it can perform protected-mode addressing to access up to 16MB of physical memory addresses. For more information, see the section titled "The 80286 Microprocessor."
22. The interrupt controller, the DMA controller, the three-channel timer/counter, and the system clock circuits. For more information, see the section titled "Integrated Peripheral Controllers (IPC)."
23. The 80486 brings an improved 80386DX microprocessor, a high-performance 80387 coprocessor, and an 82385 cache memory controller together in a single package. For more information, see the section titled "The 80486 Microprocessor."
24. The 80486 has no internal division factor to arrive at the chip's speed rating. An 80486-33 microprocessor must operate from a 33-MHz clock input signal. For more information, see the section titled "The 80486 Microprocessor."
25. Decreased size and decreased production costs. For more information, see the section titled "Chip Sets."

Chapter 7

Input/Output

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Define the overall function of the computer's input/output units.
- Describe differences between parallel and serial ports.
- Identify the various port connectors used in PC-compatible systems.
- Given a block diagram of a parallel printer interface, identify its major components.
- Define the lines of a parallel interface port.
- Describe the differences between synchronous and asynchronous transmissions, stating advantages and disadvantages for both.
- Describe the need for parallel/serial conversions.
- Describe the function and internal structure of a typical UART.
- Explain the operation of a Centronics-type parallel interface port.
- Explain the operation of an RS-232C serial interface port and define its signal lines.
- State the three tasks that must be performed by a keyboard to present meaningful data to the computer.

continues

- Describe the operation of the Universal Serial Bus (USB).
- Define the term `key bounce` and list commonly used methods of eliminating it.
- Describe various key-switching techniques used in digital computer keyboards.
- Define the term `mechanical switch`.
- Define the term `solid state switch`.
- List the events that occur when a key is depressed on the keyboard.
- Describe the operation of the PC's keyboard.
- Describe the operation of the PC's keyboard/system interface.
- Describe the operation of the keyboard controller IC.
- Describe steps to troubleshoot keyboard problems.
- State the responsibilities of an interface circuit.
- Explain the operation of a mouse and a trackball.
- Describe the operation of a game port used with joysticks and game paddles.
- Describe the operation of a light pen.
- Describe the operation of handheld and flat-bed scanners.
- Discuss optical character recognition.
- Describe steps to troubleshoot scanner problems.

Introduction

Although the circuitry on the system board forms the nucleus of the personal computer system, it cannot stand alone. The computer must be able to acquire data from the outside world. In most applications, it must also be able to deliver results of operations it performs to the outside world in a useful format. Many different systems have been developed for both inputting and outputting data.

In a PC-compatible system, more than 65,000 input and output addresses are available. Part of the preceding chapter described how the system treats its onboard, intelligent devices as I/O devices. This chapter examines peripheral I/O in detail. The first portion covers the standard I/O port assignments and configurations in PC systems. The second half of the chapter deals with typical input devices associated with PC systems.

Standard I/O Ports

Objective

Although many different methods have been developed to connect devices to the PC-compatible system, three ports have been standard since the original PCs were introduced. These are the IBM versions of the `Centronics parallel port`, the `RS-232C serial port`, and the `game port`.

Parallel Ports

Objective

Parallel ports have been a staple of the PC system since the original PCs were introduced. They are most widely used to connect printers to the computer. In many instances, parallel ports are referred to as `parallel printer ports`. Because the parallel port's capability to quickly transfer bytes of data in a parallel mode, it has been adopted to interface a number of other peripheral devices to the computer. These devices include x-y plotters; fast, computer-to-computer transfer systems; high-speed, high-volume, removable-disk backup systems; and optical scanners. The parallel port's simplicity makes it a natural for hobbyists and designers to use for specialized interface projects.

Parallel Port Operations

A simplified parallel port operation is shown in Figure 7.1. The system applies a decoded port address to the port to enable it, along with an I/O write signal. It then moves data to the port in parallel format from the system's data bus. If the device attached to the port is ready to receive data, it moves through the port still in parallel format.

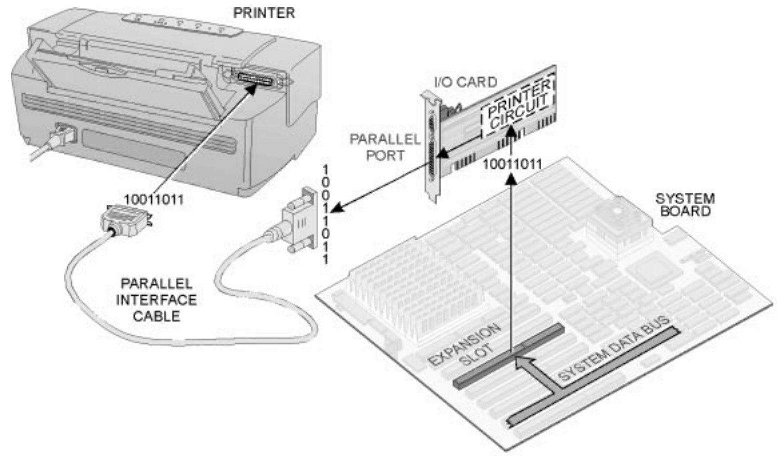


Figure 7.1

Parallel port transfers.

The port's address decoder compares each address placed on the address bus with its preset address. This address may be designated by a set of jumper wires, a bank of DIP switches, or by firmware/Plug and Play settings. All these methods allow the peripheral device's address to be changed, either by rearranging the jumpers, reconfiguring the positions of the switches, or allowing it to be detected by the BIOS' PnP firmware. Other address decoding tactics include using SSI and MSI logic circuit arrangements, or PROM devices, to decode the bits of the address bus.

When the address on the bus matches the port address, the decoder produces an enabling output (EN), which is used to enable the peripheral's sending and receiving circuits.

When a `data-out` (microprocessor-to-peripheral) operation is performed, the microprocessor places a data word on the data bus and sends an I/O write signal to the port. The port applies a `data available strobe` signal to the peripheral, indicating the presence of valid data on the bus. The strobe signal causes the data word to be transferred to the peripheral in parallel format. At the input to the peripheral, the data is latched (held) in its inbound register for processing.

When a `data-in` (peripheral-to-microprocessor) operation is performed, one of the four transmission methods previously discussed occurs to initiate the transfer of data. When the attention of the microprocessor has been gained, the peripheral's outbound register dumps its contents on the system's data bus, where it is read by the microprocessor.

The Centronics Standard

Figure 7.2 shows a typical parallel printer connection, using the IBM version of the `Centronics standard`. This interface allows the computer to pass information to the printer, eight bits at a time, across the eight data lines. The other lines in the connection carry control signals (handshaking signals) back and forth between the computer and the printer. The original Centronics interface used a 36-pin, D-shell connector at the adapter; the IBM version reduced the pin count to 25. Table 7.1 defines the signals of the IBM parallel printer port connection standard.

Figure 7.2

Parallel port signals.

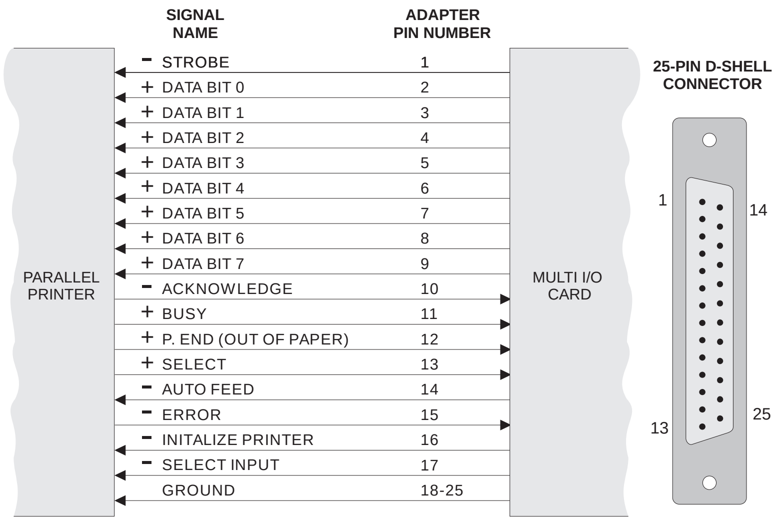


Table 7.1

Pin Assignments

Pin No.	Signal	Direction*	Description
1	STROBE	In	STROBE pulse to read data in. Pulse width must be more than 0.5µs at the printer. The signal level is normally "high"; read-in of data is performed at the "low" level.
2	DATA1	In	
3	DATA2	In	
4	DATA3	In	
5	DATA4	In	
6	DATA5	In	
7	DATA6	In	
8	DATA7	In	
9	DATA8	In	These signals represent information of the first to the eighth bits of parallel data, respectively. Each signal is at "high" level when data is logical "1"; and "low" when it is logical "0."
10	ACK	Out	Approximately 5µs pulse; "low" indicates that data has been received and the printer is ready to accept other data.
11	BUSY	Out	A "high" signal indicates that the printer cannot receive data. The signal becomes "high" in the following cases:

Pin No.	Signal	Direction*	Description
			. During data entry . During printing operation . In “Offline” state . During printer error status
12	PE	Out	A “high’ signal indicates that the printer is out of paper.
13	SLCT	Out	The signal indicates that the printer is in the selected state.
14	AUTO-FEED	In	With this signal being at “low” level, the paper is automatically fed one line after printing.
15	ERROR	Out	The level of this signal becomes “low” when the printer is in “Paper End” state, “Offline” state, or “Error” state.
16	INIT	In	When the level of this signal becomes “low,” the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at “high” level, and its pulse width must be more than 50µs at the printer.
17	SLCT IN	In	Data entry to the printer is possible only when the level of this signal is “low.”
18-25	GND	—	Ground level.

*As viewed from the printer

The data strobe line (STROBE) is used by the computer to signal the printer that a character is available on the Data lines. The printer reads the character from the data lines into its buffer, to be printed at the printer's convenience. If for some reason, the printer cannot accept the character from the data lines, such as being out of paper or its buffer being full, the printer sends a "busy" signal to the computer on the busy line, telling the computer not to send any more data.

After the peripheral device has read the data word from the data lines, it pulses the acknowledge (ACK) line to tell the computer it is ready to accept another data word, as long as the busy line is not asserted. The printer also uses the select line (SLCT) to let the computer know that data can be sent to it. In the event that the SLCT signal is not present, the computer can't send the printer any data.

In addition to the lines discussed above, the Centronics standard calls for additional printer-related control lines, which include paper end (PE), auto feed (AUTO-FD), error, initialize printer (INIT), and select input (SLCT-IN). Not all printers use the complete standard and all of its control lines. In many instances, only a few of the lines are used, and nonstandard pin numbers and connector types may be used.

Parallel Printer Interface

Objective

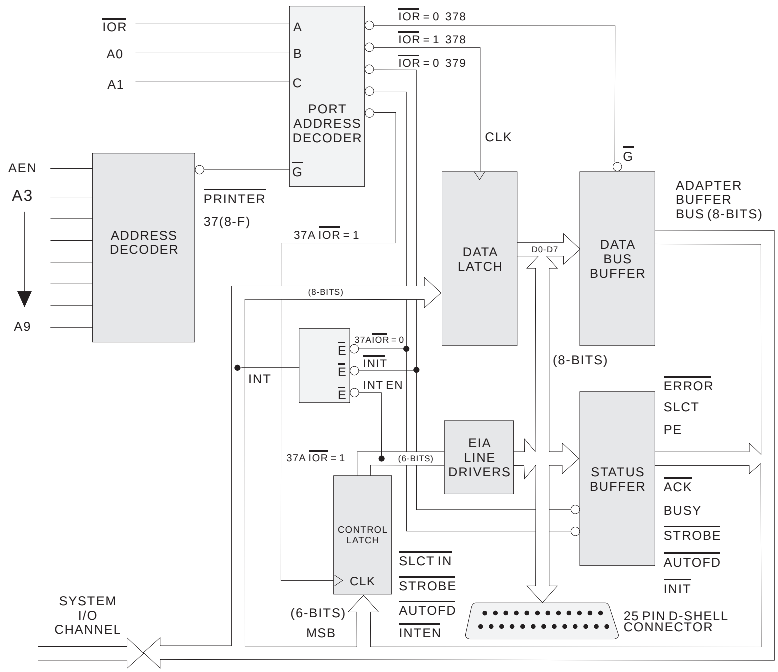
A PC-compatible parallel port interface typically offers 8-bit parallel data words and nine I/O control lines at a 25-pin, female D-shell connector at the rear of the system unit. In early PCs, the parallel port was located on the back of the video adapter card, on a dedicated parallel printer card, or on a Multi I/O card. The primary port was always located on the video adapter card. With the VGA card, the primary printer port was normally located on an MI/O card in one of the slots.

In earlier MI/O cards, all the I/O functions were controlled by discrete circuit devices. On newer MI/O cards, the various standard I/O functions were increasingly integrated into more complex ICs. Finally, IC manufacturers combined the standard MI/O functions (HDD/FDD/1P/2S/1G ports) into ASIC devices. When the Pentium system boards were developed, the ASIC chip containing the parallel port circuitry moved to the system board. The printer port connector may be located directly on the back plate of an I/O card, or the port's circuitry may be connected (via a ribbon cable) to the 25-pin, D-shell connector on the unit's back panel.

Figure 7.3 depicts a block diagram of a typical parallel printer port. The interface includes a port address decoder, data latching register, data bus buffer, control-line latching register, status-line buffer, and control-line drivers. In operation, the adapter performs five I/O instructions, which correspond to three different port addresses and the condition of the system's IOR line. The adapter supports the two-way handshaking scheme, using the strobe, acknowledge, and busy lines described earlier.

Figure 7.3

Block diagram of printer interface.



Writing to the Printer Port

The printer adapter is enabled by port addresses 378h through 37Fh. When the system wants to send a byte of data to the printer, it must place the data byte on the data bus, set its IOR line to an inactive state (this is equivalent of IOW being active to the adapter), and apply an address of 378h to the adapter. This action causes the adapter's DATA LATCH to accept the data from the data bus and latch it.

Next, the computer must read the adapter's STATUS BUFFER to check the condition of the printer's busy line. To read the printer's status bits, the computer must make its IOR line low, and apply an address of 379h to the adapter. This places the status bits (ERROR, SLCT, PE, INIT, and BUSY) on the adapter's data bus.

If the busy line is active, indicating that the printer can accept data, the system unit pulses the strobe line to tell the printer that a valid data byte is present on the data lines from the port. Actually, the system unit writes a complete, 6-bit control word into the CONTROL LATCH by placing the control word on the adapter's data bus, making the IOR line inactive, and applying an address of 37Ah to the adapter. This places the control word in the latch, which, in turn, applies the individual bits to the line driver amplifiers, and then to the printer's input control lines.

Reading from the Printer Port

Two other instructions allow the system to read the contents of the two latches. To read the current contents of the data latching register, the microprocessor must make the IOR line active and apply an address of 378h to the adapter. This action enables the data bus buffer and places the outputs of the data latch register on the adapter's data bus.

When the system unit reads the status of the printer, it activates the IOR line and applies an address of 37Ah to the adapter. This allows the system to read the printer's (SLCT-IN, ACK, STROBE, and AUTO-FD) status lines as well as the port's IRQ line. If the printer is not driving these pins, the system will read the last control word written into the control latch register. In the event that the printer is driving these pins, the status bits from the printer will be logically ORed with the bits of the control latch, and placed on the adapter's data bus.

In addition to the primary handshaking lines (STROBE, ACK, BUSY), the parallel printer interface provides secondary control lines for PE, SLCT and ERROR input signals from the printer, and AUTO-FD, INIT, and SLCT-IN outputs to the printer.

The interrupt level of the printer port section may be set at a number of different levels by changing its configuration jumpers, or CMOS enabling setting. The interrupt from the printer is actually obtained through the `initialize printer` (INIT) line. This signal is gated to the adapter's interrupt line by the `interrupt enable` (INT EN) bit of the control word stored in the control latch. The status of this bit determines whether the printer adapter can interrupt the system unit. In this manner, the printer can use the printer port's INIT input to cause an interrupt to occur, provided the system unit has not masked the interrupt through the `interrupt enable bit` of the control word. Normal interrupt settings for printer ports in a PC-compatible system are IRQ5 or IRQ7.

All the signals discussed earlier are transmitted between the adapter card and the printer at standard TTL (Transistor-Transistor Logic) levels. This means that the signals can deteriorate quickly with long lengths of cable. The cable length used for the parallel printer should be kept to less than 10 feet. If longer lengths are needed, the cable should have a low-capacitance value. The cable should also be shielded to minimize interference.

LPT Handles

DOS keeps track of the system's installed printer ports by assigning them the logical device names (handles) LPT1, LPT2, and LPT3. Whenever the system is booted up, DOS searches the hardware for parallel ports installed at hex addresses 3BCh, 378h, and 278h consecutively. If a printer port is found at 3BCh, DOS assigns it the title of LPT1. If, however, no printer port is found at 3BCh, but there is one at 378h, DOS will assign LPT1 to the latter address. Likewise, a system that has printer ports at physical addresses 378h and 278h would have LPT1 assigned at 378h, and LPT2 at location 278h.

Parallel printer ports located on video display cards are normally set for 3BCh operation. Therefore, this is the parallel port that the system will normally define as LPT1. This is also true of system boards that come equipped with built-in parallel ports. Problems will arise in any system with different parallel ports that share the same physical address. If this occurs, it will be necessary to disable, or re-address, one of the ports before either of them will work. This is usually accomplished through jumpers located on the I/O cards.

The address of the printer port can normally be changed to respond as LPT1, LPT2, or LPT3, depending on the setting of address selection jumpers. The printer port can also be disabled completely through these jumper settings. IRQ7 is normally assigned to the LPT1 printer port; IRQ5 typically serves the LPT2 port, if installed.

Although the data pins of the parallel printer port are defined as output pins, the figure illustrates that they are actually bidirectional. Many PC-compatible parallel ports are bidirectional. However, some cheaper ports may not have the electronics built into them to handle the input function. This is not important for printer operations, so most users won't notice. However, newer Enhanced Parallel Port (EPP) and Enhanced Centronic Parallel (ECP) ports can be converted between unidirectional and bidirectional operation through the CMOS setup screen. If a bidirectional port is being used to support an I/O device, such as a local area network adapter or a high capacity storage device, then this feature would need to be checked at both a hardware and software level.

Serial Ports

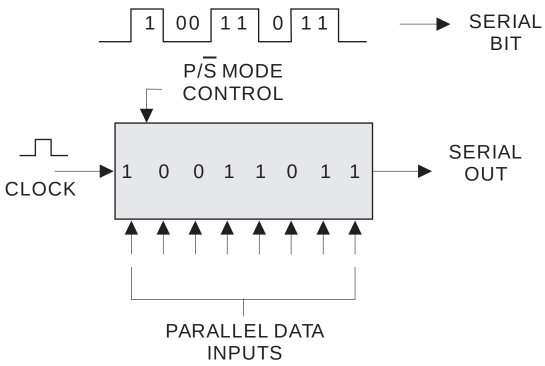
Objective

As the distance between the computer and a peripheral reaches a certain point (10 feet), it becomes less practical to send data as parallel words. An alternative method of sending data is to break the parallel words into their individual bits and transmit them, one at a time, in a serial bit stream over a single conductor. In this manner, the number of conductors connecting the computer and the peripheral is reduced from eight or more data lines, and any number of control lines, to one (or two) communications line(s), a ground line, and maybe a few control lines. Therefore, the cost of connecting equipment is reduced by using serial communication techniques when a peripheral device must be located at some distance from the computer.

The simplest method of converting a parallel computer word into a serial bit stream is depicted in Figure 7.4, using a `Parallel-In, Serial-Out` (PISO) shift register. In this arrangement, a parallel word is loaded into the register with a single clock pulse. After the word has been loaded into the register, the logic level on its mode control pin is reversed and the bits are shifted out of the register's serial output by eight consecutive clock pulses.

Figure 7.4

A PISO register.



Conversely, serial data sent from a remote computer or peripheral must be converted back into parallel form to be compatible with the computer's internal bus structure. The simplest method of implementing this operation is shown in Figure 7.5, using a `serial-in, parallel-out` (SIPO) shift register. Here, the serial bit stream is shifted into the register by eight clock pulses.

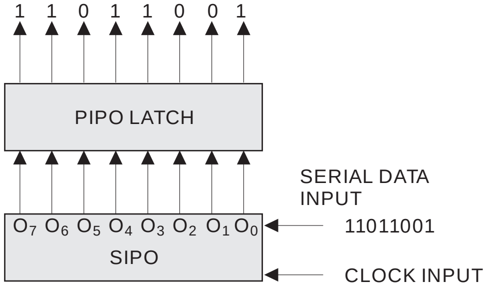
Serial Transmission Modes

The biggest problem encountered when sending data serially is keeping the transmitted data-bit timing synchronized between the two devices. Two methods are used to provide the proper timing for serial transfers: the data bits may be sent `synchronously` (in conjunction with a synchronizing clock pulse), or `asynchronously` (without an accompanying clock pulse).

Figure 7.5

A discrete SIPO register.

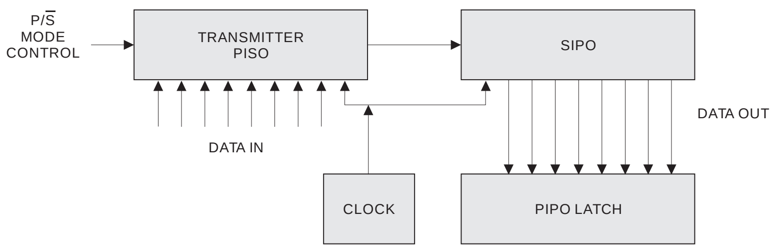
CLOCK PULSES	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
3	0	0	0	0	0	1	1	0
4	0	0	0	0	1	1	0	1
5	0	0	0	1	1	0	1	1
6	0	0	1	1	0	1	1	0
7	0	1	1	0	1	1	0	0
8	1	1	0	1	1	0	0	1



When data is transmitted synchronously, the bits of a word, or character, are synchronized by a common clock signal, which is applied to both the transmitting and receiving shift registers. The two registers are initialized before data transmission begins, when the transmitting circuitry sends a predefined bit pattern, which the receiver recognizes as the initialization command. After this, receiving circuitry processes the incoming bit stream by counting clock pulses and dividing the bit stream into words of a predetermined length. If the receiver misses a bit for any reason, all the words that follow will be processed erroneously. Figure 7.6 depicts a simplified synchronous transmission scheme.

Figure 7.6

Synchronous transmission.



When data is transferred asynchronously, the receiving system is not synchronized with the sending system. In asynchronous communications, the transmission depends on the capability of two separate clocks running at the same frequency to remain synchronized for a short period of time. The transmitted material is sent character-by-character (usually ASCII), with the beginning and end of each character framed by character start and stop bits. Between these marks, the bits of the character are sent at a constant rate, but the time interval between characters may be irregular. Figure 7.7 depicts a typical format for transmitting an 8-bit, ASCII character asynchronously.

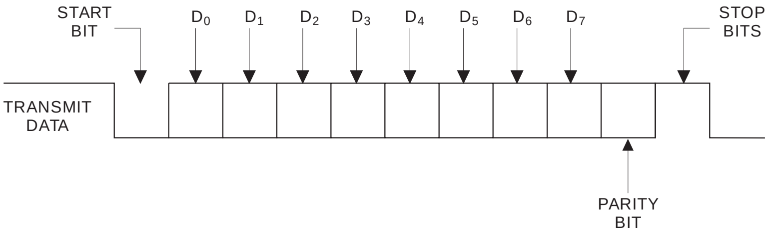


Figure 7.7

Asynchronous character format.

When no bits are being transmitted, the data line is held in a `high logic state` (mark). At the beginning of a character, the transmitter sends a `start bit`, which is always a low-logic pulse (space). After the start bit, the data bits are transmitted, beginning with the `least significant bit` (LSB). A number of bits may be transmitted after the data bits. In this case, a `parity bit` has been added for error-detection and correction purposes, and one `stop bit` (mark), which identifies the end of the character.

Although this format is fairly common, on different systems the number of data bits range between 5 and 9. There may be 1, 1-1/2, or 2 stop bits included. The use of an error-checking bit is optional.

Over a given period of time, synchronous communications are much faster than asynchronous methods. This is due to the extra number of bits required to send each character asynchronously. PC serial ports and analog modems use asynchronous communications methods; ISDN modems and local area network adapters use synchronous methods.

Serial Interface ICs

Objective

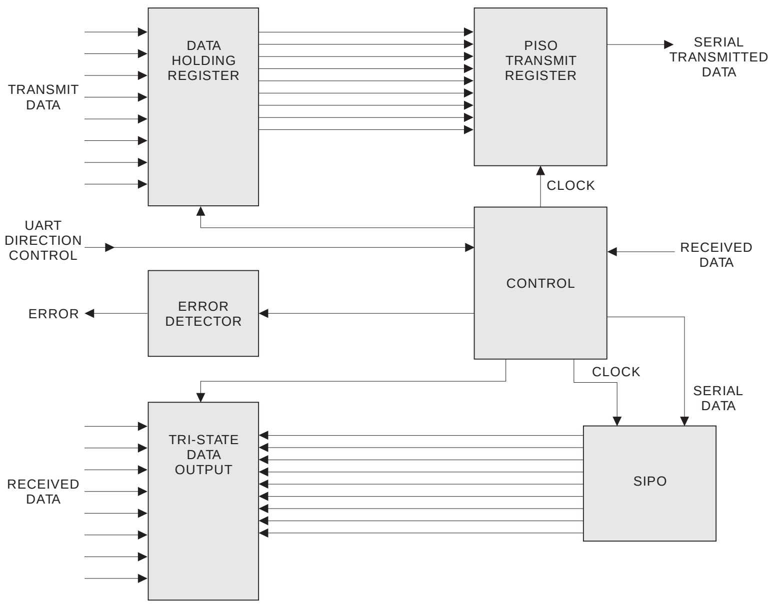
Computer systems do not normally rely on the discrete PISO and SIPO shift registers described in the previous sections. Like the single-chip parallel ports, IC manufacturers have developed a number of single-chip devices that perform all the functions necessary for serial transfers to occur. These serial port ICs are referred to as `asynchronous communication interface adapters` (ACIAs), or as `universal asynchronous receiver/transmitters` (UARTs). Synchronous devices are usually called USARTs (universal synchronous/asynchronous receiver/transmitters). Not only do these devices provide the parallel-to-serial and serial-to-parallel conversions required for serial communications, they also handle the parallel interface required with the computer's bus, and all the control functions associated with the transmission.

UARTs

A functional block diagram of a UART is illustrated in Figure 7.8. It consists of two major sections: the transmit section and the receive section. The transmit section is made up primarily of two registers: the `transmit output shift register` and a `transmit holding register`. The transmit holding register holds the next data word to be transmitted, until the shift register has completed the serialization of the preceding data word.

Figure 7.8

Blocks of a UART.



The UART's receiver section is basically the reverse of the transmit section. Serial data is shifted into the serial receive shift register until the predetermined number of bits has been accumulated. At this point, the bits are loaded (in parallel form) into the receive holding register. Both the transmit and receive functions are under the direction of the UART's control section. Many of the device's parameters, such as the number of start and stop bits and the type of parity (if any) to be used during transmission, may be programmed by the user through the control section.

In pre-Pentium units, the system's Multi I/O card provided a pair of fully programmable, asynchronous communication channels through two serial port connections. On earlier I/O cards, a pair of 8250 UARTs were used as the basic port circuitry. In newer MI/O cards, a single VLSI device, called an integrated I/O controller, provides the interfacing and UART functions. In most Pentium systems, the serial port adapter function is incorporated into the system board's integrated I/O controller IC.

The original serial adapters featured programmable baud rates from 50 to 9600 baud, a fully programmable interrupt system, and variable character lengths (5-, 6-, 7-, or 8-bit characters). In addition, the adapter added and removed start, stop, and parity bits, had false start-bit detection, line-break detection and generation, and possessed built-in diagnostics capabilities. As modems became faster and faster, upgraded UARTs were included, or integrated, to keep up.

Notable advanced UART versions include the 16450 and 16550. The 16450 was the 16-bit improvement of the 8250; the 16550 was a high-performance UART, with an onboard 16-byte buffer. The buffer allows the UART to store or transmit a string of data without interrupting the system's microprocessor to handle them. This provides the 16550 with an impressive speed advantage over previous UARTs. These advanced UARTs allow serial ports to reach data-transmission rates of up to 115 Kbps. Although some features have changed between these UARTs, and although they are sometimes combined directly into an integrated I/O chip, they must still adhere to the basic 8250 structure to remain PC-compatible.

USARTs

As their name implies, USARTs have the capability to perform asynchronous communications like the UART, but when higher-performance data transfers are required, the USART is used in synchronous mode. A few words of clarification about why synchronous transmission would offer higher performance (speed) than asynchronous transmission may be in order at this point. Recall that for asynchronous communications a start bit is required to mark the beginning of each character, and at least one stop bit is required to identify the end of the character. This means that at least 10 bits (and therefore 10-bit times) are required to send an 8-bit data character.

On the other hand, synchronous transmissions require no non-data bits after the transmitter and receiver have been initialized. Therefore, 8 bits of data are transmitted in 8-bit times. The only drawback is that any break in the data stream will cause the transmitter and receiver to become desynchronized. To minimize this problem, the USART incorporates extra internal circuitry to generate "dummy" or "null" characters for transmission. This keeps the data flow-rate in sync when no actual data is being sent. In addition, the USART also incorporates a set of `first-in, first-out` (FIFO) registers. These registers can be filled in advance (this is called `queuing`) so that a constant flow of data, and therefore, synchronization, can be maintained between the transmitter and receiver.

Serial Interface Connections

Objective

Because of the popularity of asynchronous serial data transmissions and the number of devices that use them, such as printers and modems, standardized bit-serial signals and connection schemes have been developed to simplify the connecting of serial devices to computers. The most popular of these serial interface standards is the `Electronic Industry Association` (EIA) RS-232C interface standard.

The RS-232C Standard

Basically, the IBM version of the RS-232C standard calls for a 25-pin, male D-type connector. It also designates certain pins for data transmission and receiving, along with a number of control lines. Table 7.2 identifies these pin designations. The standard was developed to cover a wide variety of peripheral devices, and therefore not all the lines are used in any given application. Different device manufacturers may use different combinations of RS-232C lines, even between peripherals of the same type.

Table 7-2

RS-232C Signal Lines

Pin No.	Common Name	RS-232C Name	Description
1		AA	Protective ground
2	TxD	BA	Transmitted data
3	RxD	BB	Received data
4	RTS	CA	Request to send
5	CTS	CB	Clear to send
6	DSR	CC	Data set ready
7	GND	AB	Signal ground (common return)
8	CD	CF	Received line signal detector (RLSD)
9			Reserved for data set testing
10			Reserved for data set testing
11			Unassigned

continues

Table 7-2 Continued

Pin No.	Common Name	RS-232C Name	Description
12	SI	SCF	Secondary received line signal detector
13		SCB	Secondary clear to send
14		SBA	Secondary transmitted data
15		DB	Transmission signal element timing (DCE source)
16		SBB	Secondary received data
17		DD	Received signal element timing (DCE source)
18			Unassigned
19		SCA	Secondary request to send
20	DTR	CD	Data terminal ready
21		CG	Signal quality detector
22	RI	CE	Ring indicator
23		CH/CI	Data signal rate selector (DTE/DCE source)
24		DA	Transmit signal element timing (DTE source)
25			Unassigned

In addition to defining the type of connector to be used and the use of its individual pins, the RS-232 standard also establishes acceptable voltage levels for the signals on the pins. A logic "1" is represented by a voltage between –3 and –20 volts DC. Conversely, a logic "0" is signified by a voltage between +3 and +20 volts DC. These levels are generally converted to and from TTL-compatible signals by CMOS driver and receiver chips. Under these conditions, a maximum rate of 20,000 baud can be achieved for distances under 50 feet.

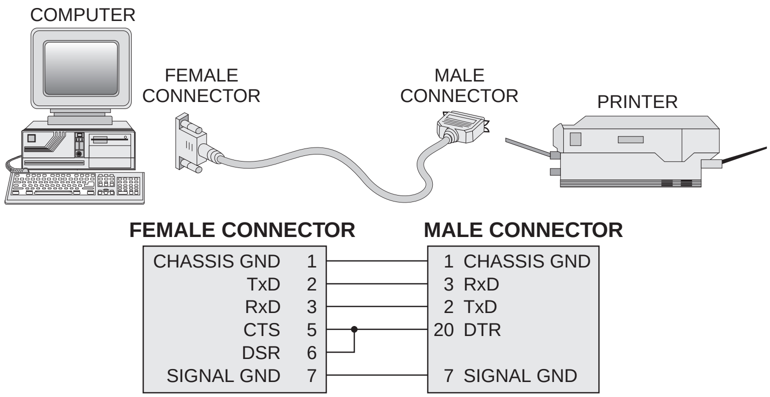
Advanced Serial Standards

Since the adoption of the RS-232C standard, the EIA has also adopted two more improved serial standards: the RS-422 and RS-423, which are enhancements of the RS-232C standard. The RS-422 uses twisted-pair transmission lines, and differential line signals to provide a high degree of noise immunity for transmitted data. The RS-423 standard uses coaxial cable to provide extended transmission distances and higher data transfer rates.

With the advent of the mouse as a common input device, a 9-pin, male D-shell version of the RS-232 serial port became common. This version is commonly used as the COM1 serial port for the mouse, in Windows-based systems. Figure 7.9 depicts the 9-pin version of the interface being used to connect a serial printer through a common connection scheme referred to as a "null modem."

Figure 7.9

RS-232C 9-pin serial printer connection.



The exchanging of pins 2 and 3 between the two devices forms the basis of the null modem. Because the device in Figure 7.10 is a serial printer, pins 5 and 6 of the DTE equipment are tied to the DTR pin of the DCE equipment. In a true null modem, pins 4 and 5 of the interface would need to be cross-connected to facilitate two-way communications.

The character bit stream is transmitted to the printer on the line designated as the `transmit data` line (TXDATA) at the computer connector and `receive data` line (RXDATA) at the printer connector. A reciprocal line (TXDATA at the printer connector, and RXDATA at the computer connector) is also used in the printer interface. Because data does not flow from the printer to the computer, this line basically informs the computer that a printer is connected to the interface, turned on, and ready to receive data (much like the select line in the Centronics interface standard).

The flow of data to the printer is moderated by a line referred to as the `data set ready` (DSR) line at the computer connector, and `data terminal ready` (DTR) line at the printer connector. The printer uses this line in much the same manner as the busy line of the Centronics interface. When the buffer is full, the printer signals on this line to tell the computer to not send any more data. More complex serial interfacing may include a line called the `clear to send` (CTS) line at the computer connector and `ready to send` (RTS) line at the printer connector, and its reciprocal line, where the identifications are reversed.

At the printer's end of the cable, another UART receives the serial bit stream, removes the start and stop bits, checks the parity bit for transmission errors, and reassembles the character data into parallel form.

Because the movement of data is asynchronous using the UART, an agreement must be established between the computer's UART and the printer's UART concerning the speed at which characters will be sent. The transmission, or `baud`, rate of the computer's UART is generally set by software. On the other hand, the printer's baud rate is usually designated by a set of DIP switches in the printer. Common baud rates used with serial printers are 300, 1,200, 2,400, and 9,600 `bits per second` (bps). One of the most common problems associated with getting a serial interface to work is mismatched baud rate.

Character Framing

In the Centronics-type parallel port, the data is specifically sent in individual 8-bit packages. In serial communications, however, there is more flexibility in how the data is transmitted. Typical RS-232 transmission formats allow 7- or 8-bit characters to be sent as a package inside of a `character frame`. Each frame also contains various numbers of non-data bits for marking the start and stopping points of the frame. An additional bit is often added to the frame for error-checking purposes. The composition of the character frame must be the same at both the sending and receiving ends of the transmission. It shouldn't be too difficult to understand what problems would arise if a device is set to receive a 7-bit character, with two start bits, one stop bit, and an error-checking bit; but receives an 8-bit character with two start bits, two stop bits, and no error-checking bit.

Using the serial printer as an example, it should be easy to envision simple communications taking place. However, most serial port applications involve two-way communications. In these types of applications, control of the communication port becomes more complicated. In addition to matching baud rates and character framing between the two devices, a mechanism for controlling the flow of information between the two ports must be established. This method of controlling the flow of information between the two devices is called a `protocol` and must be agreed to by both devices for the transfer to be successful. Both hardware and software flow control protocols have been devised for use with serial ports. More information about character framing and protocols is presented in Chapter 11, "Data Communications."

DOS Serial Port Names

As with parallel ports, DOS assigns COM port designations to the system's serial ports during bootup. COM port designations are normally COM1 and COM2 in most systems, but they can be extended to COM3 and COM4 in advanced systems.

Either RS-232 port may be designated as COM1, COM2, COM3, or COM4, as long as both ports are not assigned to the same COM port number. In most PCs, COM1 is assigned as port address hex 3F8h and use IRQ channel 4. The COM2 port is typically assigned port address hex 2F8h and IRQ3. Likewise, COM3 uses IRQ4 and is assigned an I/O address of 3E8h, while COM4 usually resides at 2E8 and uses IRQ4.

Game Port

The game control adapter allows up to four game paddles or two joysticks to be used with the system. The adapter converts resistive input values into relative paddle or joystick positions, in much the same manner described in the preceding section. This adapter can also function as a general purpose I/O converter, featuring four analog and four digital input points. Figure 7.10 depicts a block diagram of the game control adapter.

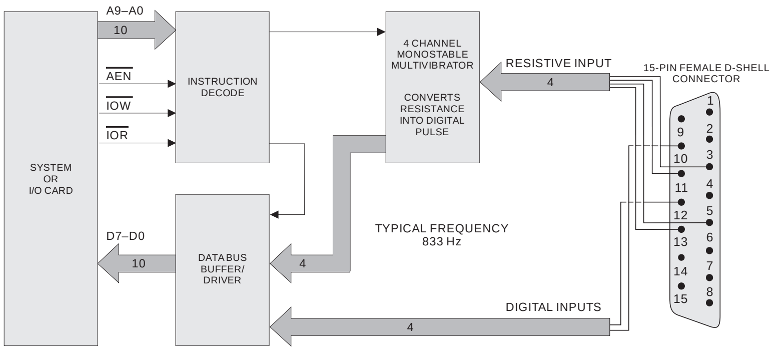


Figure 7.10

The game control adapter.

The input to the game port is generally a set of resistive joysticks or game paddles. Joysticks are defined as having two variable resistances, and game paddles as consisting of one variable resistance. Each resistance should be variable between 0 and 100 k-ohms. Joysticks may have one or two normally-open (fire) buttons. Game paddles have only one button. Under this definition, the game control adapter will support two game paddles (A and B), four game paddles (A, B, C, and D), or two joysticks (A and B). The order of fire buttons should correspond with that of the resistive elements (A and B or A, B, C, and D).

This interface is simple and straightforward, combining elementary hardware and software techniques. When the software issues the adapter's address (201h) and an IOW signal, the timers are triggered into their active state. The individual timer outputs remain active for a length of time, dictated by the setting of each resistive input and timing capacitor at each timer's input. As each timer times out according to its own resistive/capacitive (RC) time constant, its output returns to a nonactive state.

The software periodically polls the adapter's latching register data to determine whether each output has timed out. A software counter keeps track of the number of times the port has been read before the timers time out. The number of read cycles (0 to 255) before a logic low is encountered is directly proportional to the resistive setting of the joystick or game paddle.

Troubleshooting Port Problems

Three levels of testing basically apply to troubleshooting port problems. These are the `DOS level`, the `Windows level`, and the `hardware level`. Before concentrating on any of these levels, troubleshooting should begin by observing the symptoms produced by operation of the port. Observe the steps that lead to the failure. Determine under what conditions the port failed. Was a new peripheral device installed in the system? Were any unusual operations in progress? Note any error messages or beep codes. Use the troubleshooting hints that follow to isolate the parallel, serial, or game port circuitry as the source of the problem. Retry the system several times to observe the symptoms clearly. Take the time to document the problem.

Figure 7.11 illustrates the components involved in the operation of the serial, parallel, and game ports. Failures in these devices tend to end with poor or no operation of the peripheral. Generally, there are only four possible causes for a problem with a device connected to an I/O port: the port is defective, the software is not configured properly for the port, the connecting signal cable is bad, or the attached device is not functional.

Port Problem Symptoms

Typical symptoms associated with serial, parallel, or game port failures include the following:

- A 199, 432, or 90x IBM-compatible error code is displayed on the monitor (printer port).
- Online light is on, but no characters are printed by the printer.

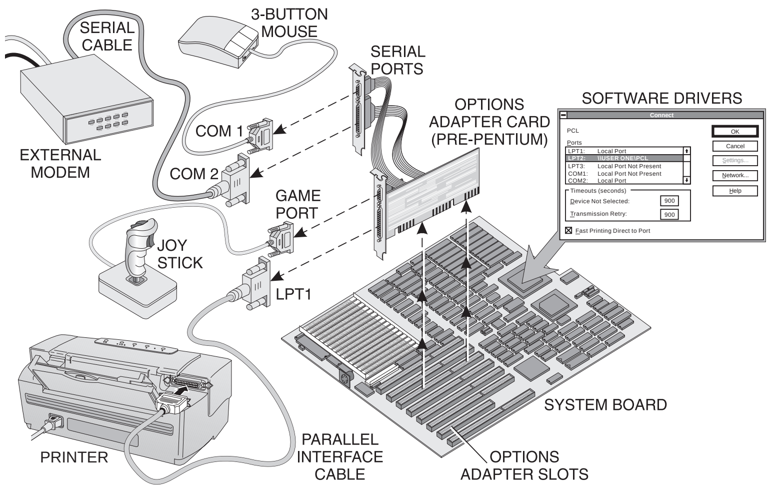


Figure 7.11

Components associated with I/O ports.

- A 110x IBM-compatible error code is displayed on the monitor (serial port).
- Device-not-found error message or unreliable connection.
- Input device will not work on game port.

As you can see from the symptoms list, I/O ports do not tend to generate many error messages onscreen.

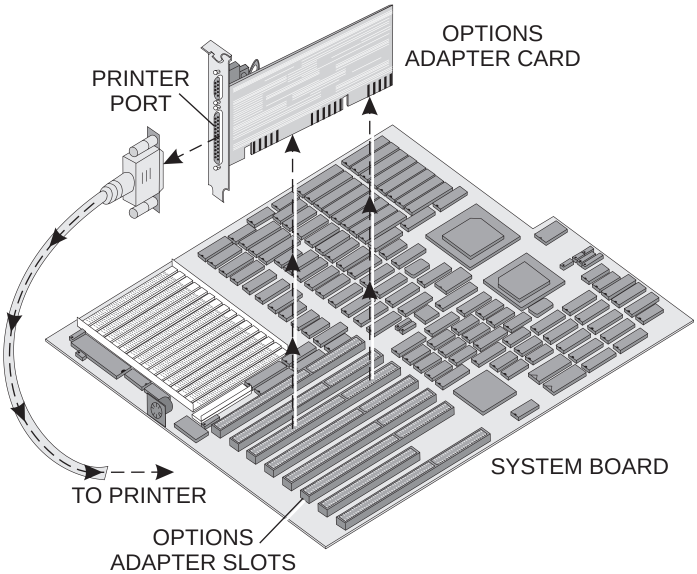
Port Hardware Checks

In the area of hardware, only a few items pertain to the system's ports—the port connector, signal cabling between the port circuitry and the connector in some units, the port circuitry itself, and the system board. As mentioned earlier, the port circuitry can be found on video cards in some older units, on specialized I/O cards in other units, and on the system board in newer units. In any of these situations, some configuration settings must be correct. Check the board containing the I/O port circuitry and its User's Guide, for configuration information. This will normally involve LPT, COM, and IRQ settings. Occasionally, you will be required to set up hexadecimal addressing for the port addresses; however, this is becoming rare.

With newer Pentium systems, it will be necessary to check the Advanced CMOS Setup to determine whether the port in question has been enabled, and, if so, has it been enabled correctly. For example, a modern parallel port must be enabled and set to the proper protocol type in order to operate advanced peripherals. For typical printer operations, the setting can normally be set to *Standard Printer Port* (SPP) mode. However, devices that use the port in a bi-directional manner will need to be set to EPP or ECP mode for proper operation. In both cases, the protocol must be set properly for both the port and the device in order to carry out communications.Figure 7.12 illustrates the movement of data to the parallel port mounted on an I/O adapter card.

Figure 7.12

Moving information through a port.



It is helpful to single-step through the bootup in order to read the port assignments in the bootup window. If serial or parallel port problems are occurring, the CMOS configuration window is the first place to look. If the system does not detect the presence of the port hardware at this stage, none of the more advanced levels will find it either. If values for any of the physical ports installed in the system do not appear in this window, check for improper port configuration jumpers or switches.

Because the unit has not loaded DOS at the time the configuration window appears, DOS and Windows cannot be sources of port problems at this time. If all jumpers and configuration settings for the ports appear correct, assume that a hardware problem exists. Diagnose the hardware port problem to a section of the system (in this case the board containing the port).

DOS Checks

If possible, run a software diagnostics package to narrow the possible problem causes. This is not normally a problem because port failures do not generally affect the main components of the system. Try to use a diagnostic program that deals with the system board components. It should include parallel- and serial-port tests, as well as a game-port test if possible.

Run the program's `Port Tests` function and perform the equivalent of the ALL Tests function. Note all the errors indicated by the tests. If a hardware error is indicated, such as those already mentioned, it may be possible to take some corrective actions, such as resetting or reconfiguring LPT, COM, or IRQ settings, without replacing the unit containing the port hardware. If more complex port problems are indicated, exit the diagnostic program and replace the port hardware.

DOS Parallel Ports

Software diagnostic packages normally ask you to place a `loop-back test plug` in the parallel port connector to run tests on the port. The loop back plugs simulate a printer device by redirecting output signals from the port into port input pins. Figure 7.13 depicts the signal rerouting scheme used in a parallel port loop-back plug.

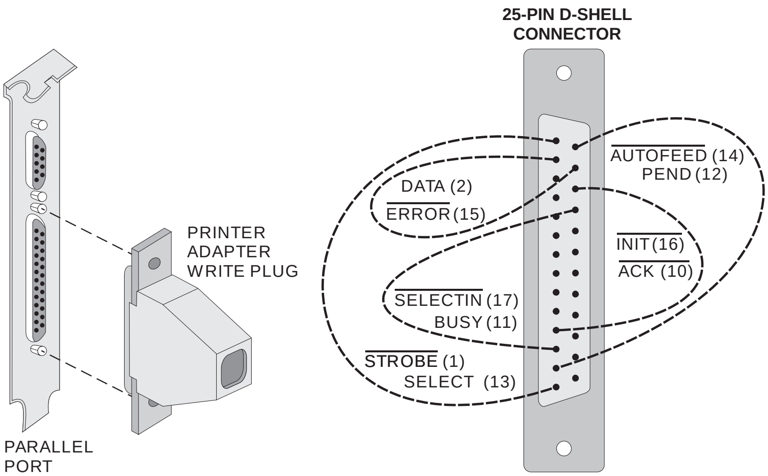


Figure 7.13

Parallel port loop-back connections.

A live printer can be used with the port for testing purposes, but this action elevates the possibility that problems can be injected into the troubleshooting process by the printer.

If the software diagnostic program does not provide enough information to solve the problem, attempt to print to the first parallel port from the DOS level. To do this, type **COPY AUTOEXEC.BAT LPT1:** at the DOS prompt and press the Enter key.

If the file is not successfully printed, type **EDIT AUTOEXEC.BAT** at the C:\ >> DOS prompt. Check the file for a `SET TEMP =` command. If the command is not present, add a `SET TEMP` statement to the AUTOEXEC.BAT file. At the C:\ >> DOS prompt, type **EDIT AUTOEXEC.BAT**. Create a blank line in the file and type **SET TEMP=C:\WINDOWS\TEMP** into it. Save the updated file to disk and reboot the system. Make sure to check the `SET TEMP=` line for blank spaces at the end of the line.

Is there a printer switch box between the computer and the printer? If so, remove the print sharing equipment, connect the computer directly to the printer, and try to print from the DOS level as previously described.

Check the free space on the HDD. Remove any unnecessary files to clear space on the HDD, and defragment the drive as described in Chapter 13, "Preventative Maintenance and Safety."

DOS Serial Ports

As with parallel ports, diagnostic packages typically ask you to place a loop back test plug in the serial port connector in order to run tests on the port. Use the diagnostic program to determine whether there are any IRQ or addressing conflicts between the serial port and other installed options. The serial loop back plug is connected differently (physically) than a parallel loop back plug so that it can simulate the operation of a serial device.

A live serial device can be used with the port for testing purposes but, like the printer, this elevates the possibility that non-port problems can be injected into the troubleshooting process.

If the software diagnostic program does not provide enough information to solve the problem, attempt to print to the serial port from the DOS level. To do this, type **DIR. COM*x*** at the DOS prompt. The value of x is equal to the COM port being printed to.

Windows 3.x Checks

Windows adds another level of complexity to isolating port problems. Windows was designed to perform the printer and communication functions for the system. If DOS-level port operation is working, but not Windows, check under the Control Panel icon in the Main window. There is no Game Port setting in Windows. Therefore this port can only be tested from the DOS level.

The `[Ports]` entry of the WIN.INI file is used to define parallel printer and serial ports. Up to 10 ports may be defined in this section: four serial ports, three parallel ports, two logical ports, and a file port. The SYSTEM.INI device-driver files control and communicate with the port hardware.

Windows 3.x Parallel Ports

Try to print from a non-Windows environment. Return to the DOS prompt and open a non-Windows application that has a print function. Attempt to print from the application.

If the system will print from DOS, but not from Windows, determine whether the Print option from the application's File menu is unavailable (gray). If so, check the Windows Control Panel/Printers window for correct parallel port settings. Make certain that the correct printer driver is selected for the printer being used. If no printer, or the wrong printer type is selected, just set the desired printer as the Default printer. To add the desired printer as the Default printer, enter the Main window, double-click on the Control Panel icon, double-click on the Printer icon, and set the desired printer as the Default printer.

Click on the Setup button to examine the selected printer's settings. If these settings are correct, click on the Connect button to ensure that the printer information is being routed to the correct port. This sequence is depicted in Figure 7.14.

The final place to check for Windows printer-port problems is in the Print Manager. This area of Windows is reached through the Main window by double-clicking on the Print Manager icon. Check the Print Manager for errors that have occurred which may be holding up the printing of jobs that follow. If an error is hanging up the print function, highlight the offending job and remove it from the print spool by clicking on the Delete Document entry of the Document menu. These steps are summarized in Figure 7.15.

Figure 7.14

Printers and Connect windows.

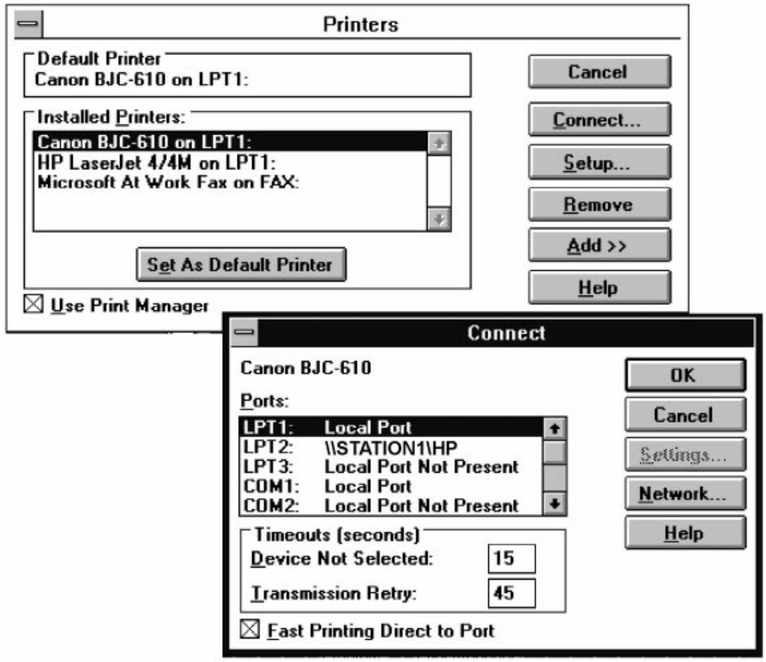
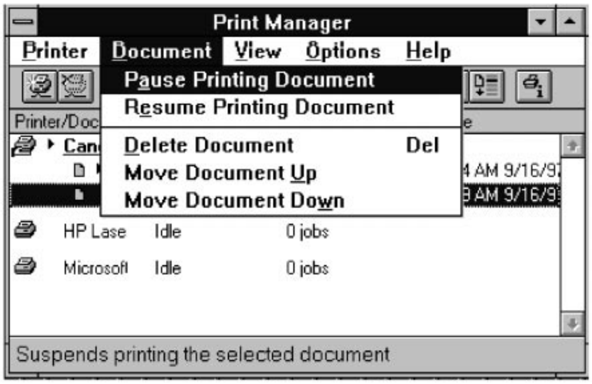


Figure 7.15

Print Manager window.



Windows 3.x Serial Ports

Check the Windows Control Panel/Ports window for correct serial port settings. This is accomplished by double-clicking on the Main tile. Double-click on the Control Panel icon to access it, click on the Ports option, and click on the Settings button. Most serial printers use settings of `9600 Baud`, `No Parity`, `8 Bits`, `1 Stop Bit`, and `Hardware Handshaking`. Click on the Advanced button to determine the IRQ setup for the port. Check the User's Manual to document the correct settings for the device using the port in question.

It is possible to check the internal setup of the Windows serial ports. From the DOS editor, check the SYSTEM.INI file for COM port configuration information. At the DOS prompt, type **Edit C:\WINDOWS\SYSTEM.INI**. Scroll down to the `[386enh]` section and check the COM Port values. In Windows, mice are normally installed on COM1 or COM2. IRQ4 is normally assigned to COM1 and COM3; IRQ3 is usually designated for COM2 and COM4.

Windows 95 Checks

The I/O port functions in Windows 95 can be reached through two avenues. Port information can be accessed through the desktop's Start/Settings buttons. This information can also be reached through the My Computer icon on the desktop. Printer port information can be viewed through the Printers icon, and serial port information is accessed through the System/Device Manager entries under the Control Panel icon.

Windows 95 Parallel Ports

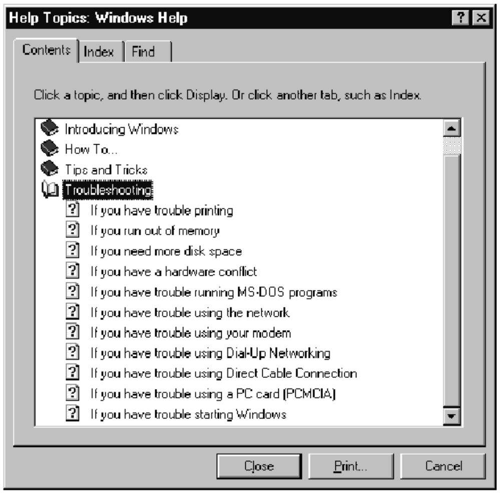
Isolate the problem to the Win95 program by attempting to print from a non-Windows environment. Restart the computer in MS-DOS mode, and attempt to print a batch file from the DOS prompt.

If the system will print from DOS, but not from Win95, check to see whether the Print option from application's File menu is unavailable (gray). If so, check the My Computer/Printers window for correct parallel port settings. Make certain that the correct printer driver is selected for the printer being used. If no printer or the wrong printer type is selected, use the Add Printer Wizard to install and set up the desired printer.

Windows 95 comes with an online tool, called Print Trouble-shooter, to help solve printing problems. To use the Print Troubleshooter, click the Troubleshooting entry in the Win95 Help system, as illustrated in Figure 7.16. Press F1 to enter the Help system. The Troubleshooter will ask a series of questions about the printing setup. After all of its questions have been answered, the Troubleshooter returns a list of recommendations for fixing the problem.

Figure 7.16

Accessing Windows 95 Troubleshooting Help.



If the conclusions of the Troubleshooter do not clear up the problem, try printing a document to a file. This will enable you to separate the printing software from the port hardware. If the document successfully prints to a file, use the DOS copy command to copy the file to the printer port. The format for doing this is `Copy /b filename.prn lpt1:`. If the document prints to the file, but will not print out on the printer, the hardware setup and circuitry are causing the problem.

Continue troubleshooting the port by checking the printer driver to ensure that it is the correct driver and version number. Click on the Printer icon and select the Properties entry from the menu. Click on the Details tab to view the driver's name. Click on the About entry under the Device Options tab to verify the driver's version number.

Click on the printer port in question (under the Printer icon) to open the Print Manager screen. Check the Print Manager for errors that have occurred that may be holding up the printing of jobs that follow it. If an error is hanging up the print function, highlight the offending job and remove it from the print spool by clicking on the Delete Document entry of the Document menu.

Windows 95 Serial Ports

Information on the system's serial ports is contained in three areas under the Device Manager. These are the Resources entry, the Driver entry, and the Port Settings entry. The Resources entry displays port-address ranges and IRQ assignments. The Driver entry displays the names of the installed device drivers and their locations. The Port Settings entry, depicted in Figure 7.17, contains speed and character frame information for the serial ports. The Advanced entry under Port Settings allows the transmit and receive buffer speeds to be adjusted for better operation.

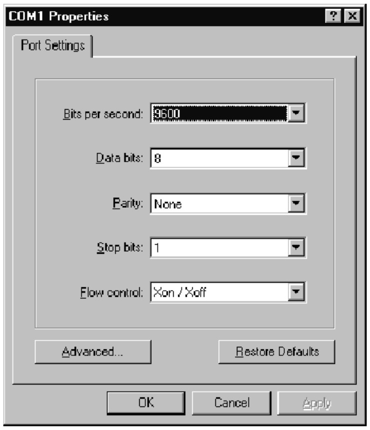


Figure 7.17

Port Settings entry.

Check under the Win95 Control Panel/System/Device Manager window for correct serial port settings. Click on the Port Settings option to see the setup for the ports. Most serial printers use settings of `9,600 Baud`, `No Parity`, `8 Bits`, `1 Stop Bit`, and `Hardware Handshaking` (XON-XOFF). Click on the Resources button to determine the IRQ setup for the port. Check the User's Manual to document the correct settings for the device using the port in question.

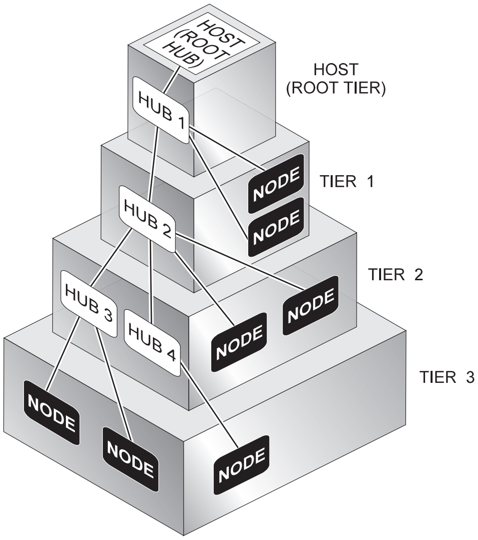
Universal Serial Bus (USB)

Objective

A new serial interface scheme, called the `Universal Serial Bus`, has been developed to provide a fast, flexible method of attaching up to 127 peripheral devices to the computer. The USB provides a connection format designed to replace the system's traditional serial and parallel port connections. USB peripherals can be daisy-chained, or networked together using connection hubs that allow the bus to branch out through additional port connections. The resulting connection architecture is a tiered-star configuration, like the one shown in Figure 7.18.

Figure 7.18

Universal Serial Bus architecture.



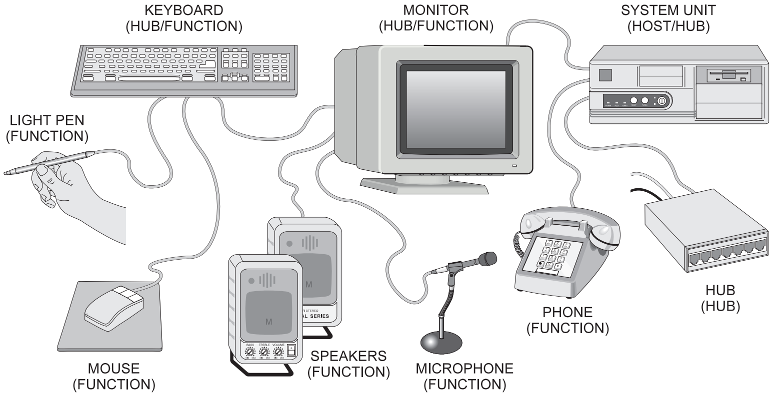
The USB system is composed of a `USB host` and `USB devices`. The devices category consists of `hubs` and `nodes`. In any system, there is one USB host. This unit contains the interface that provides the USB `host controller`. The controller is actually a combination of USB hardware, firmware, and software.

Hubs are devices that provide additional connection points for other USB devices. A special hub, called the `root hub`, is an integral part of the host system and provides one or more attachment points for USB devices.

Although the tiered architecture shown in Figure 7.19 approaches the complexity and capabilities of the local area networks covered in Chapter 11, "Data Communications," a more practical desktop connection scheme is presented in Figure 7.20. It is evident that some of the components of the system serve as both a function and a hub (for example, the keyboard and monitor). In these devices, the package holds the components of the function, and also provides an embedded hub that other functions can be connected to. These devices are referred to as `compound devices`.

Figure 7.19

USB desktop connection scheme.

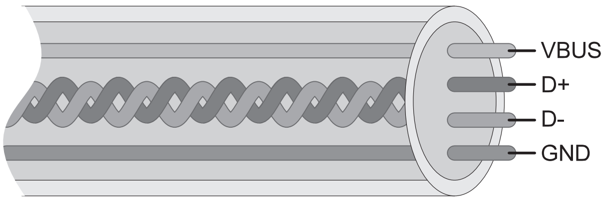


USB Cabling and Connectors

USB transfers are conducted over a four-wire cable, as illustrated in Figure 7.20. The signal travels over a pair of twisted wires (D+ and D–) in a 90 ohm cable. The differential signal and twisted-pair wiring provide minimum signal deterioration over distances and high noise immunity.

Figure 7.20

The USB cable.



A Vbus and ground wire are also present. The Vbus is the +5 Vdc power cable. The interface provides power to the peripheral attached to it. The root hub provides power straight from the host system to those devices directly connected to it. Hubs also supply power to the devices connected to them. Although the interface supplies power to the devices, they are allowed to have their own power sources, if necessary. In these instances, the device must be designed specifically to avoid interference with the bus's power distribution scheme. The USB host's power-management software can apply power to devices when needed and suspend power to it when not required.

The USB specification defines two types of plugs: series-A and series-B. Series-A connectors are used for devices where the USB cable connection is permanently attached to devices at one end. Examples of these devices include keyboards, mice, and hubs. Conversely, the series-B plugs and jacks are designed for devices such as printers, scanners, and modems, that require detachable cabling. Both are four-contact plugs and sockets embedded in plastic connectors, as illustrated in Figure 7.21. The sockets can be implemented in vertical, right-angle, and panel-mount variations. The icon used to represent a USB connector is also depicted in the figure.

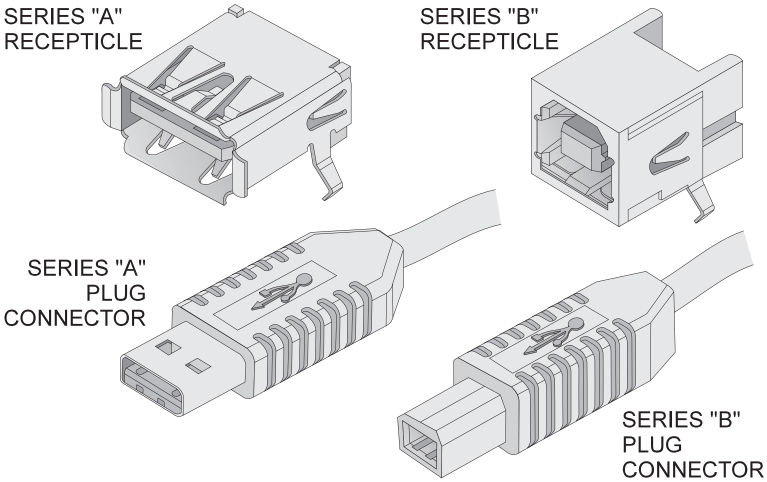


Figure 7.21
USB connectors.

The connectors for both series are keyed so that they cannot be plugged in backward. All hubs and functions possess a single, permanently attached cable with a series-B connector at its end. The connectors are designed so that the A and B series connections cannot be interchanged.

USB Data Transfers

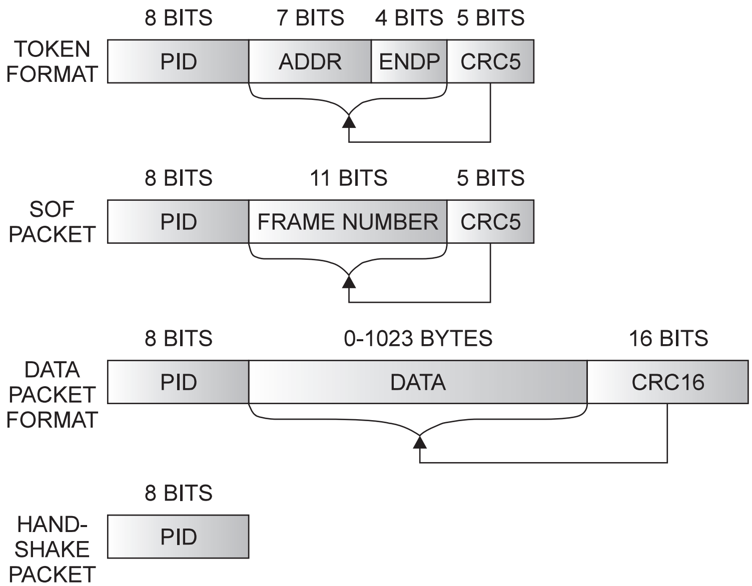
Unlike traditional serial interfaces that transmit framed characters one at a time, data moves across the USB in the form of `data packets`. Packet sizes vary with the type of transmission being carried out. However, they are typically 8, 16, 32, or 64 bytes in length. All transmissions require that two or three packets of information be exchanged between the host, the source location, and the destination location.

All data transfers are conducted between the host and an endpoint device. The flow of data can occur in either direction. USB transactions begin when the host controller sends a token packet that contains information about the type of transaction to take place, the direction of the transmission, the address of the designated USB device, and an endpoint number. If the device is the source of the transaction, it either places a data packet on the bus or informs the host that it has no data to send. If the host is the source, it just places the data packet on the bus. In either case, the destination returns a handshake packet if the transfer was successful. If an error is detected in the transfer, a not acknowledge (NACK) packet is generated. Figure 7.22 demonstrates the USB's four packet formats. They are the token packet, the start-of-frame (SOF) packet, the data packet, and the handshake packet.

Each type of packet begins with an 8-bit packet ID (PID) section. The SOF packet adds an 11-bit frame number section and a 5-bit cyclic redundancy check (CRC) error checking code section. In the data packet a variable-length data section replaces the frame number section, and the CRC frame is enlarged to 16 bits. The data section can range up to 1,023 bytes in length. The handshake packet consists of just a PID byte.

Figure 7.22

USB packet formats.



The USB management software dynamically tracks which devices are attached to the bus and where they are. This process of identifying and numbering bus devices is known as `bus enumerating`. The USB specification allows `hot-swap` peripheral connection that does not require the system to be shut down. The system automatically detects peripherals and configures the proper driver. Instead of just detecting and charting devices at startup in a PnP style, the USB continuously monitors the bus and updates the list whenever a device is added to or removed from it.

The USB specification allows for the following types of transfers to be conducted:

- `Control transfers` are used by the system to configure devices at startup or time of connection. Other software can use control transfers to perform other device-specific operations.
- `Bulk data transfers` are used to service devices, such as scanners and printers, that can handle large batches of data. Bulk transfers are typically made up of large bursts of sequential data. The system arranges for bulk transfers to be conducted when the bus has plenty of capacity to carry out the transfer.
- `Interrupt transfers` are small, spontaneous transfers from a device that are used to announce events, provide input coordinate information, or transfer characters.
- `Isochronous transfers` involve large streams of data. This format is used to move continuous, real-time data streams such as voice or video. Data delivery rates are predetermined and correspond to the sampling rate of the device.

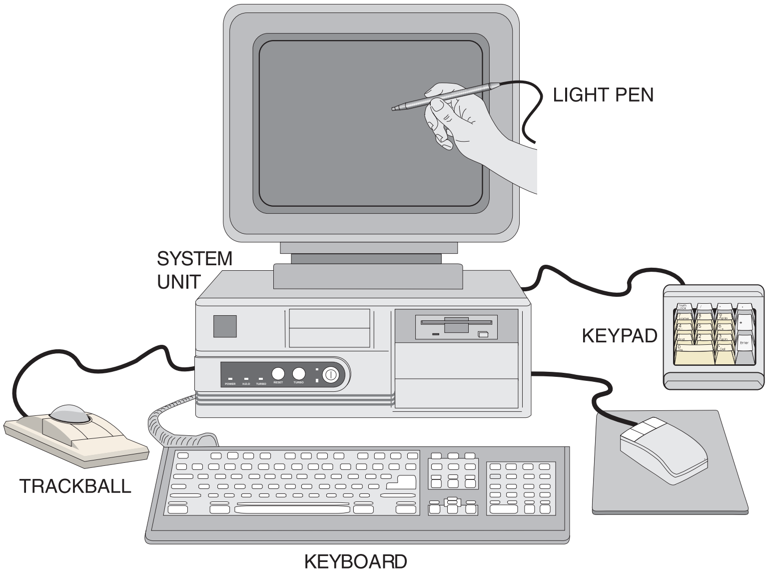
Input Devices

Input devices convert physical quantities into electronic signals that can be manipulated by interface units. The input devices typically used with microcomputers convert human physical activity into electronic impulses that can be processed by the computer. The chief devices of this type are keyboards, joysticks, mice, light pens and touch-panel screens. These devices are illustrated in

Figure 7.23. Other types of input devices convert physical quantities, such as temperature, pressure, and motion into processable signals. These devices are normally found in industrial-control applications.

Figure 7.23

Input devices.



Keyboards and Keypads

Objective

The `alphanumeric keyboard` is the most widely used input device for microcomputers. It provides a simple, finger-operated method of entering numbers, letters, symbols, and special-control characters into the computer. Modern computer keyboards are actually adaptations of earlier typewriter-like keyboards used with teletypewriters. These devices served as both input and output systems for the earliest generation of computers. In addition to the alphabetic and numeric keys found on conventional typewriters, the computer keyboard may also contain any number of special function and command keys to extend the keyboard's basic operation and provide special-purpose entry functions.

In smaller microcomputers, such as those programmed in hexadecimal code, numeric-only `keypads` may be used as the sole means of entering programs and data. In some computer keyboards, numeric keypads, resembling calculator keypads, are included with the alphanumeric keys. When large amounts of numeric data must be entered into the computer, the keypad usually proves more efficient than the numeric keys associated with the alphanumeric portion of most keyboards.

Keyboard Designs

The pattern in which keyboards are arranged and constructed has traditionally sparked some debate among users. Everyone seems to have a favorite key pattern and "feel" he or she prefers in a keyboard. Obviously, an individual who is trained to touch-type on a standard QWERTY typewriter keyboard would prefer that the computer keyboard be laid out in the same manner. The contour of the key top, the amount of pressure, and the length of the stroke that must be applied to the key to actuate it are also important ergonomic considerations in keyboard design. Some keyboards offer a defined `click` at the bottom of the keystroke to identify a complete entry. Others offer shorter keystrokes with a soft bottom and no feedback click.

Keyboard design is part form and part function. Although the QWERTY keyboard remains the standard for key arrangement, a second key pattern—known as the DVORAK keyboard—has gained some notoriety. This keyboard layout, depicted in Figure 7.24, attempts to arrange the keyboard characters in a more logical pattern that should lead to faster operation. Its basic premise is that the new key placement should lead to alternate hand usage.

Although the layout is quite old (devised in 1936) and does offer some speed advantages, it has only gained minimal acceptance by users. However, the programmability of PC-compatible keyboards make DVORAK conversion easy if desired. The only physical action required to implement a DVORAK layout involves repositioning of the keyboard's keycaps.

Still other designs have been adapted to make using the keyboard more comfortable for the user. Some units include a special cushion along the front edge to provide support for the wrists. Another innovative design divides the keyboard in half and angles each side back slightly, as illustrated in Figure 7.25. This design is supposed to offer a more natural angle for the human hands than straight-across designs.

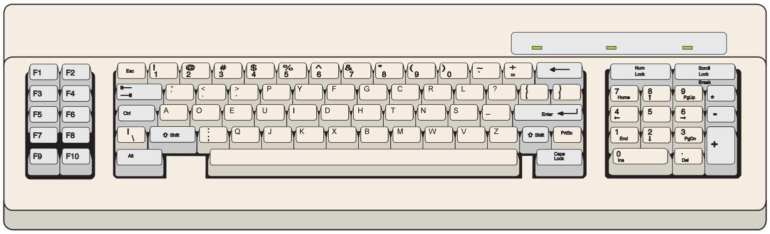


Figure 7.24

A DVORAK keyboard.

Figure 7.25

An ergonomic keyboard design.

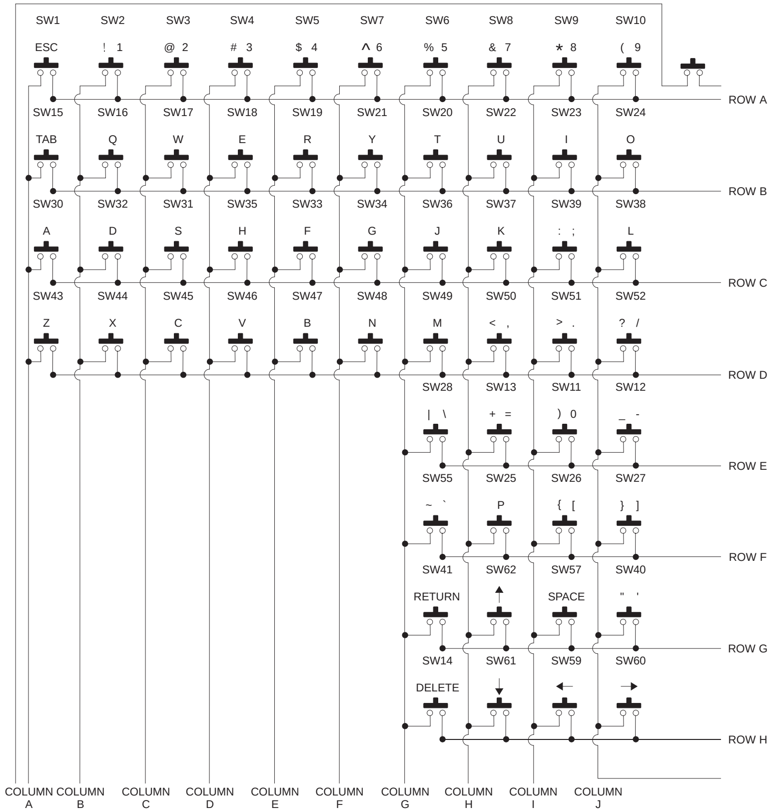
Keyboard Encoding

Inside, a keyboard is basically an `x-y matrix` arrangement of switch elements, as shown in Figure 7.26. To produce meaningful data from a key depression, the keyboard must be able to detect and identify the depressed key, debounce it, and encode the key closure into a form usable by the computer.

Depressing one of the keys `shorts` a row to a column. The closure is detected by applying an active logic level to the rows, either one at a time, or collectively, and scanning the columns for an active level. When an active logic level is detected on a column, the row/column intersection represents the key closure.

Figure 7.26

A typical keyboard matrix.



After the detection and identification process has been completed, a short delay is imposed and the key is rechecked. A `time delay` is one method of eliminating a characteristic associated with all types of keyboard switches, known as `key bounce`. Key bounce occurs just prior to the switch closure, and just after its release, while the switch contacts are in close proximity to each other. During these periods (5–20 milliseconds), contact-arcing can produce pulses that may be erroneously interpreted as additional digital signals.

To transform the coordinates of the key closure into a more conventional code—such as hex, EBCDIC, or ASCII—the row and column information is applied as an address to a ROM look-up table. This table holds the corresponding character codes. The same function can also be performed by a software routine.

As a matter of fact, all the functions discussed so far can be performed with software routines. Software interfacing has the advantages of lower cost, and less board space than would be required to perform the same tasks with hardware. The disadvantage of software interfacing is the fact that the system microprocessor must expend a great deal of time involved in keyboard operations.

On the other hand, advancements in LSI chip technology have produced single-device keyboard encoders that contain all the circuitry necessary to interface the computer to an electronic keyboard. By allowing hardware to handle the keyboard functions, the system's microprocessor is freed up to perform other, more important functions. The system's microprocessor needs to be involved only with the keyboard long enough to transfer the character from the keyboard after it has been notified that a valid key closure has occurred. Figure 7.27 depicts a block diagram of a typical hardware keyboard encoder.

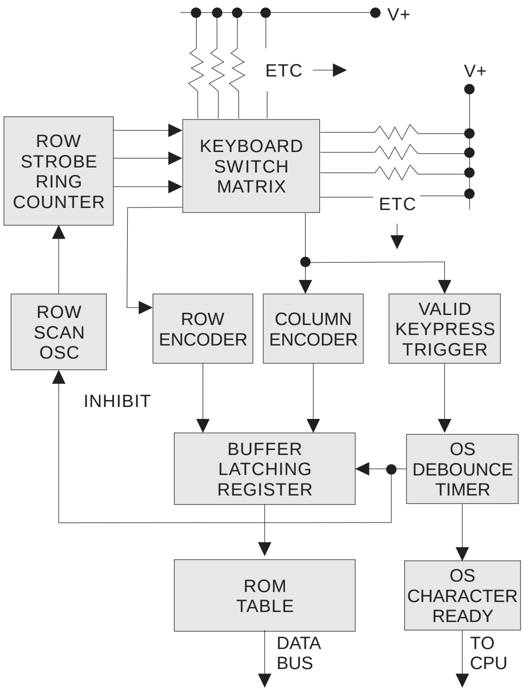


Figure 7.27

A keyboard encoding scheme.

A row-scan oscillator is used to clock a row strobe ring counter. The counter circulates an active output level sequentially to the rows of the matrix. When a key is depressed, and the row strobe places the active level on that row, an active level is produced at the corresponding column. The high-to-low transition on any of the column lines triggers the one-shot debounce timer. The row and column logic patterns are combined through two encoders to form an address for the ROM look-up table.

The debounce timer performs three functions. When a key closure is detected, the timer output inhibits the scan oscillator to prevent further scanning until the closure encoding is completed. Second, the debounce timer clocks the scan-code latch to allow the combined row/column address code to be applied to the ROM table. Finally, the debounce timer triggers a second one-shot, which produces a character-ready signal to inform the system that a valid character has been entered.

A more advanced hardware encoding method uses a microcontroller (dedicated microprocessor) to monitor the keyboard matrix, as shown in Figure 7.28. The microcontroller scans the matrix and generates an interrupt request signal when a key closure occurs. The microcontroller performs all the functions previously discussed and offers greater flexibility in operation. This includes the capability to evaluate multiple, simultaneous key closures for legitimate entries, simplified multiple-key entries (control characters, lowercase letters, and so forth) faster scan rates, and the capability to redefine the function of various keys.

Switching Techniques

Objective

Several methods of switching are used to perform the keying function in computer keyboards. The oldest of these is the simple mechanical switch, depicted in Figure 7.29. It functions in much the same manner as a common light switch. Although mechanical switches have been the mainstay of switching technology for some time, techniques using electronic parameters, such as capacitance and inductance, have gained widespread acceptance.

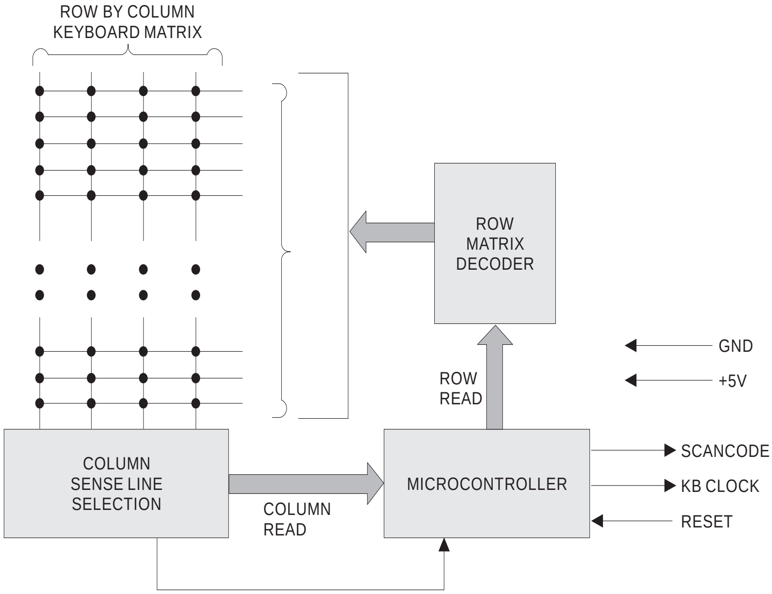


Figure 7.28

Keyboard encoding with a controller.

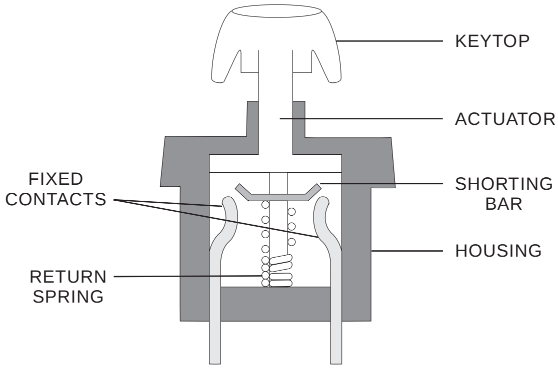


Figure 7.29

A hard-contact switch.

Mechanical Switches

The term `mechanical switch`, as it applies to computer keyboards, actually covers a fairly wide range of mechanical devices. Mechanical keyboards can be separated into two basic groups:

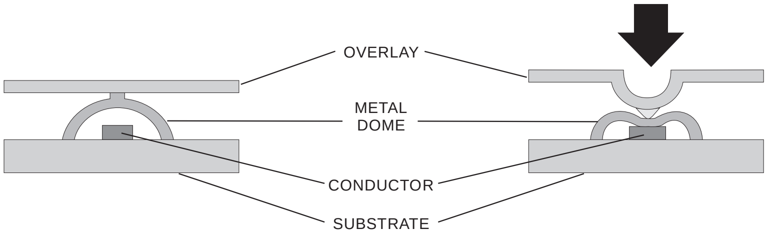
- The hard-contact keyboard, where individual switches provide the keys, and
- Key panels that are constructed as integrated units

The switch in the preceding figure depicts a typical `hard-contact key switch.` These switches are widely used in keyboards because they are relatively inexpensive. They do, however, tend to suffer from excessive key bounce and a relatively short lifespan.

A somewhat newer mechanical switching technique is the `snap-action dome switch`, shown in Figure 7.30. A small stainless-steel dome is connected to a conductor on a PC board. A second conductor passes beneath the dome. When pressure is applied to the top of the dome by a key plunger, or by touching a plastic overlay, contact is made between the dome and the center conductor.

Figure 7.30

A snap-action dome switch.



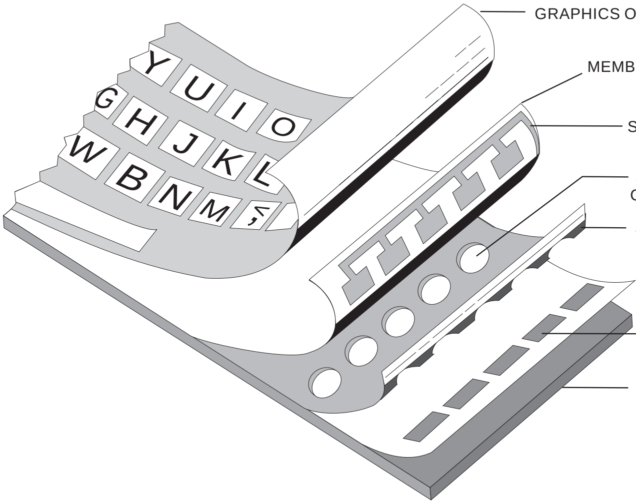
When pressure is released, the dome snaps back to its original shape and contact between the dome and the conductor is removed.

The snap-action dome switch may be used as with key caps, where the individual switches are set in a keyboard arrangement. They may also be used in a dome switch field, covered by a common overlay. This is referred to as a key panel arrangement. Another key panel technology is the simple `membrane key panel`, depicted in Figure 7.31.

The membrane key panel is a flexible, three-layer lamination whose upper and lower layers are etched with adjacent metallic contact points. The separating layer holds the contact points apart under normal circumstances. However, it has openings between the contact points to allow contact to occur under pressure. The membrane is normally attached to a rigid substrate to prevent unintentional flexing of the laminations, causing the contact points to erroneously meet. The entire assembly is covered by a plastic overlay containing graphic images that identify, and mark, the location of the contact points.

Figure 7.31

A membrane keyboard.

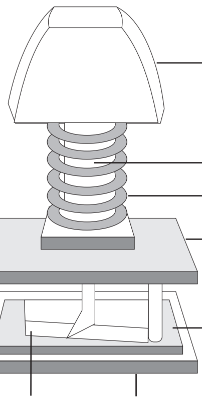


Solid State Switches

New types of non-contact electronic switching techniques have gained wide acceptance in computer keyboards. This is due to their low cost, high performance, and longevity of operation. Two of these methods are the `capacitive switching` and the `magnetic-core switching` techniques. Capacity switching is depicted in Figure 7.32.

Figure 7.32

A capacitive switch.



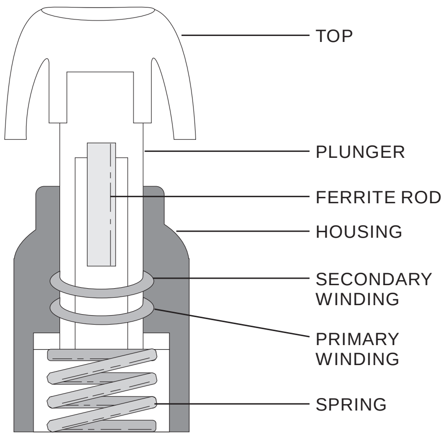
`Capacitive switches` work by using the varying capacitance between two switch contacts to change the frequency of an oscillator. The change in oscillator frequency is detected by the keyboard encoder and converted to a key closure signal.

The switch works on the most basic principle of a capacitor. A capacitor is just two conductive plates separated by a dielectric (insulator, such as air). Its capacitance is directly proportional to the area of its plates, and inversely proportional to the distance between them. In the switch, a movable plate approaches a fixed plate when the key is depressed. The capacitance of the switch increases as the plates approach each other, and the oscillator's output frequency decreases. The capacitive key switch is easily the most popular technology used in keyboards today.

`Magnetic-core switches` use a ferro-magnetic keystroke plunger to alter the magnetic coupling between the two coils of a small transformer, as illustrated in Figure 7.33. When the plunger is in its nondepressed state, the magnetic core is out of the transformer, and its windings are loosely coupled. Depression of the key moves the core into the transformer, increasing its coupling. This allows pulses applied to the primary coil to be induced into the secondary coil with enough amplitude to be recognized as a valid output signal.

Figure 7.33

A magnetic-core switch.



Interfacing and Connection

Objective

Most early microcomputers included the keyboard as an integral part of the system unit. Most manufacturers now, however, produce detachable keyboards that connect to the computer by a flexible cable. This allows the user greater movement potential and therefore, increased comfort.

Note

Extending the detachable keyboard concept one step further, the IBM PCjr offered completely cordless operation, by transferring keyboard entries to the system over an infrared link.

Most detachable keyboards use a round, 1/2-inch, 5-pin DIN connector to plug into the PC's system board. The connection is most often made through a round opening in the rear of the system unit's case. In some case designs, a front-mounted, 5-pin plug-in is included. The front-mounted connector is routed to the system board through an extension cable.

With the IBM PS/2 line, a smaller (1/4-inch) 6-pin mini-DIN connector was adopted. Other PC-compatibles use a modular, 6-pin AMP connector to interface the keyboard to the system. Figure 7.34 shows the various connection schemes used with detachable keyboards.

Figure 7.34

Connection schemes for detachable keyboards.

Key	Make/Break Code	Key	Make/Break Code	Key	Make/Break Code	Key	Make/Break Code
Esc	01 / 81	J	24 / A4	' / ~	29 / A9	7 / Home	47 / C7
1 / !	02 / 82	K	25 / A5	Lft Shift	2A / AA	8	48 / C8
2 / @	03 / 83	L	26 / A6	\/ |	2B / AB	9 / PgUp	49 / C9
3 / #	04 / 84	M	32 / B2	,/	33 / B3	-	4A / CA
4 / $	05 / 85	N	31 / B1	./	34 / B4	4	4B / CB
5 / %	06 / 86	O	18 / 98	// ?	35 / B5	5	4C / CC
6 / ^	07 / 87	P	19 / 99	Rt Shift	36 / B6	6	4D / CD
7 / &	08 / 88	Q	10 / 90	^	37 / B7	+	4E / CE
8 / ^	09 / 89	R	13 / 93	Lft Alt	38 / B8	1 / End	4F / CF
9 / (	0A / 8A	S	1F / 9F	Rt Alt	E0 38 / E0 B8	2	50 / E0
0 /)	0B / 8B	T	14 / 94	Space	39 / B9	3 / PgDn	51 / E1
- / _	0C / 8C	U	16 / 96	Caps Lock	3A / BA	0 / Ins	52 /E2
= / +	0D / 8D	V	2F / AF	F1	3B / BB	./ Del	53 / E3
Backspace	0E / 8E	W	11 / 91	F2	3C / BC	F11	57 / D7
Tab	0F/ 8F	X	2D / AD	F3	3D / BD	F12	58 / D8
A	1E / 9E	Y	15 / 95	F4	3E / BE	Up Arrow	E0 48 / E0 C8
B	30 / B0	Z	2C / AC	F5	3F / BF	Dn Arrow	E0 50 / E0 D0
C	2E / AE	[/ {	1A/9A	F6	40 / C0	Lft Arrow	E0 4B / E0 CB
D	20 / A0	] / }	1B / 9B	F7	41 / C1	Rt Arrow	E0 4D / E0 CD
E	12 / 92	Enter	1C / 9C	F8	42 / C2	Home	E0 47 / E0 C7
F	21 / A1	Lft Ctrl	1D / 9D	F9	43 / C3	End	E0 4F / E0 CF
G	22 / A2	Rt Ctrl	E0 1D / E0 9D	F10	44 / C4	Ins	E0 52 / E0 D2
H	23 / A3	;/:	27 / A7	Num Lock	45 / C5	Del	E0 53 / E0 D3
I	17 / 97	'/"	28 / A8	Scroll Lock	46 / C6	PgUp	E0 49 / E0 C9
						PgDn	E0 51 / E0 D1

PC-Compatible Keyboards

Objective

Figure 7.35 is a diagram of a typical, 101-key keyboard that comes with most PC systems. It contains an onboard microprocessor, which provides fast scan rates and the capability to evaluate multiple key closures. The `keyboard encoder` is actually an 8-bit microprocessor, specifically adapted for `intelligent` keyboard operations. It has both internal RAM (128×8) and ROM (2kb × 8), along with specialized electronics to handle keyboard control functions. The keyboard manufacturer programs the encoder to handle the particular keyboard being manufactured.

The system's keyboard-handling routines provide a `keyboard buffer` memory, which allows up to 10 characters to be typed ahead of the present operation. It also allows the user to redefine the keys in any manner desired, via software routines. Transfers of data from the keyboard to the system unit are performed serially.

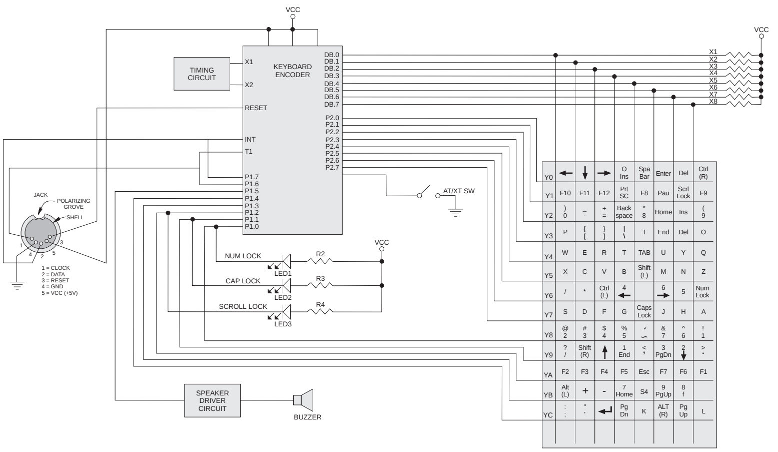


Figure 7.35

A 101-key keyboard.

The keys of this PC's keyboard are arranged in a matrix of 13 active-low STROBE lines and 8 normally high SENSE lines. The keyboard encoder `scans` the lines of the matrix by sequentially dropping each of its scan lines to a low-logic level. This causes each row of the matrix to be active for a short duration. The keyboard encoder sets the row scan rate much faster than it is humanly possible to close one of the key switches and release it. The encoder scans the entire keyboard within 3 to 5 milliseconds.

When a switch closure shorts a particular row to a particular column, a low-logic level appears at one of the keyboard encoder's sense inputs. When the active-logic level is detected, the keyboard encoder pauses for a few milliseconds to allow the switch closure to settle out. In this manner, the keyboard encoder provides its own switch-debouncing, without using any additional hardware. After this short delay, the keyboard encoder stores the closure in its buffer and continues scanning until all the rows have been scanned.

At the end of the scan, the status of the entire keyboard has been stored in the encoder's internal buffer. This allows it to evaluate the keyboard for phantom switch closures and legitimate, multiple key closures (such as Ctrl+Alt+Del). When the encoder detects two or more switches closed in the same row, it concludes that a phantom closure has occurred and generally ignores the condition completely.

Each time the keyboard encoder receives a valid key closure from the matrix, it generates two serially coded characters. The first is a distinct 8-bit code (referred to as a scan code) that corresponds to the key closure. The second is a break code that is generated when the key closure is broken. The break code is the same as the scan code, with the exception that the most significant bit of the break code is high.

The keyboard encoder's scan codes are listed in Figure 7.36. These codes are transmitted to the PC's system unit on the scan code data line (KBDATA). When no code is being sent, the keyboard encoder holds this line high. If a key (or set of keys other than the Shift key) is held down for more than half a second, the keyboard encoder will begin regenerating the same scan code, at a rate of 10 times per second, until key-break is detected.

The encoder notifies the system unit that it is ready to transmit a scan code by dropping the serial data line to a low-logic level for 0.2 millisecond. This forms a (start-bit). It then begins transmitting the code, least-significant bit first, with each bit time defined as 0.1 millisecond in length. To coordinate the serial bit stream with the receiving circuitry, the keyboard encoder sends a keyboard clock (KBCLK) signal to the system unit.

Figure 7.36

Keyboard scan codes.

Key	Make/Break Code	Key	Make/Break Code	Key	Make/Break Code	Key	Make/Break Code
Esc	01 / 81	J	24 / A4	' / ~	29 / A9	7 / Home	47 / C7
1 / !	02 / 82	K	25 / A5	Lft Shift	2A / AA	8	48 / C8
2 / @	03 / 83	L	26 / A6	\/ |	2B / AB	9 / PgUp	49 / C9
3 / #	04 / 84	M	32 / B2	, /	33 / B3	-	4A / CA
4 / $	05 / 85	N	31 / B1	. /	34 / B4	4	4B / CB
5 / %	06 / 86	O	18 / 98	// ?	35 / B5	5	4C / CC
6 / ^	07 / 87	P	19 / 99	Rt Shift	36 / B6	6	4D / CD
7 / &	08 / 88	Q	10 / 90	*	37 / B7	+	4E / CE
8 / *	09 / 89	R	13 / 93	Lft Alt	38 / B8	1 / End	4F / CF
9 / (	0A / 8A	S	1F / 9F	Rt Alt	E0 38 / E0 B8	2	50 / E0
0 /)	0B / 8B	T	14 / 94	Space	39 / B9	3 / PgDn	51 / E1
- / _	0C / 8C	U	16 / 96	Caps Lock	3A / BA	0 / Ins	52 / E2
= / +	0D / 8D	V	2F / AF	F1	3B / BB	. / Del	53 / E3
Backspace	0E / 8E	W	11 / 91	F2	3C / BC	F11	57 / D7
Tab	0F / 8F	X	2D / AD	F3	3D / BD	F12	58 / D8
A	1E / 9E	Y	15 / 95	F4	3E / BE	Up Arrow	E0 48 / E0 C8
B	30 / B0	Z	2C / AC	F5	3F / BF	Dn Arrow	E0 50 / E0 D0
C	2E / AE	[/ {	1A/9A	F6	40 / C0	Lft Arrow	E0 4B / E0 CB
D	20 / A0	] / }	1B / 9B	F7	41 / C1	Rt Arrow	E0 4D / E0 CD
E	12 / 92	Enter	1C / 9C	F8	42 / C2	Home	E0 47 / E0 C7
F	21 / A1	Lft Ctrl	1D / 9D	F9	43 / C3	End	E0 4F / E0 CF
G	22 / A2	Rt Ctrl	E0 1D / E0 9D	F10	44 / C4	Ins	E0 52 / E0 D2
H	23 / A3	; / :	27 / A7	Num Lock	45 / C5	Del	E0 53 / E0 D3
I	17 / 97	' / "	28 / A8	Scroll Lock	46 / C6	PgUp	E0 49 / E0 C9
						PgDn	E0 51 / E0 D1

Keyboard Interfacing

Objective

On PC and XT system boards, the keyboard interfacing function was handled by discrete logic circuits. The interfacing was based on a 74LS322 SIPO shift register. The incoming data bits from the keyboard were shifted into the register serially. A special two-pulse delay circuit in the data line caused the register to ignore the two start bits at the beginning of the incoming data stream. When the eight bits of the register became full, the shift register IC produced the system's IRQ1 signal, and placed its bits on the system's data bus.

Intelligent Keyboard Controllers

On many AT-compatible system boards, a single-chip microcontroller performs the keyboard interfacing functions. This specialized microcontroller is called an intelligent keyboard controller. It contains built-in control software, stored in an internal ROM area, that allows its operation to be customized for keyboard control. When the keyboard controller receives serial data from the keyboard, it checks the parity of the data, converts it into a scan code (which it stores in its output buffer), and generates an interrupt (IRQ1) to the system.

Figure 7.37 depicts the internal block diagram of an 8042/8742 intelligent keyboard controller. This type of device has been widely used in AT-compatible systems. From its internal structure, it should be easy to see that it is indeed a microprocessor, with an arithmetic logic unit (ALU), onboard control circuitry, and a wealth of internal registers. This structure gives hardware and software designers a great deal of flexibility in the design and use of the computer's keyboard interface. For most purposes, however, the greatest points of interest in the keyboard controller are `input and output data bus buffers` (DBBIN and DBBOUT), its `input port register` (P1.0–P1.7), `output port register` (P2.0–P2.7), and its `test lines` (T0–T1).

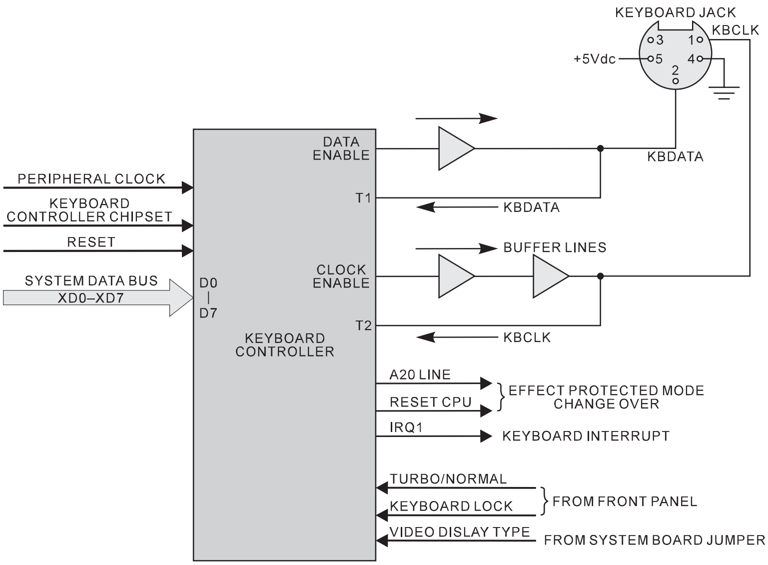


Figure 7.37

A typical keyboard controller IC.

The keyboard interface allows data to be sent serially to the keyboard. This is accomplished by writing the data into the controller's input buffer. The data byte written into this location can be transmitted to the keyboard (with an odd parity bit added for error-detection purposes). Afterwards, the controller will wait for an acknowledge signal from the keyboard before allowing another byte to be written to the keyboard.

Consider the typical system board's keyboard controller circuitry. The interface circuitry supports a round, 5-pin DIN socket compatible with the keyboard plug found on nearly all keyboards manufactured for use with compatibles. Keyboard data passes through pin 2 of the connector (KBDATA); clocking signals move through pin 1 (KBCLK). A keyboard reset, which can be connected to the system's RESET line, is included at pin 3. Ground and +5 Vdc connections are applied to the keyboard through pins 4 and 5 respectively.

The front panel connections of most PCs have a BERG connector that allows an external keylock switch to be used to control access to the computer. If this switch is set to the Lock position when the system performs its initialization routine, the bootup program will halt without reaching the DOS prompt.

The Input Buffer

The microcontroller's 8-bit input buffer serves two functions, depending on how it is addressed. At hex address 60h, data written into the buffer is interpreted as data to be transmitted to the keyboard. However, data written to this buffer using address 64h will be handled as a command to the controller.

The system's software writes to this buffer to manipulate the keyboard controller's output port register. Through this register, software can enable or disable the keyboard data and keyboard clock lines (KBEN and KBCLKEN). Other bits of this register cause the system to move back and forth between real and protected addressing modes, and change the system's operating speed between normal and turbo modes. The interface is constructed in such a way that when the system sends commands to the keyboard, it uses the KBEN and KBCLKEN lines as its data and clock outputs to the keyboard. Table 7.3 defines the bits of the keyboard controller's output port (P2.0–P2.7).

Table 7-3

The Keyboard Controller's Input- and Output-Port Bit Definitions

Input Port Bit	Definition	Output Port Bit	Definition
0	RESERVED	0	CPU RESET 0=CPU RESET 1=CPU NOT RESET
1	RESERVED	1	GATE ADDRESS BUT A20 0=A20 INHIBITED 1=A20 LINE NOT INHIBITED
2	RESERVED	2	RESERVED
3	SOFTWARE TURBO	3	SOFTWARE TURBO SWITCH
4	SWITCH	4	OUTPUT BUFFER FULL (IRQ1) 0=BUFFER NOT FULL 1=BUFFER FULL (IRQ1)
5	RESERVED	5	RESERVED
6	RESERVED	6	KEYBOARD CLOCK (OUT) KBCLKEN 0=KB CLOCK ENABLED 1=KB CLOCK DISABLED
7	KEYBOARD INHIBIT 0=KEYBOARD INHIBITED 1=KEYBOARD UNLOCKED	7	KEYBOARD DATA (OUT) KBEN

Real/Protected Mode Switching

The A20 bit of the address bus is used when the system changes over from real to protected addressing mode and back. To switch into protected mode, the system software must do the following:

- Enable the system's A20 address line
- Clear all interrupts
- Store the address to that the processor should return when real mode is re-entered
- Load the microprocessor's protected-mode tracking registers
- Unmask the interrupt handler routines
- Set the microprocessor's protected-mode enable (PE) bit

When the microprocessor is first powered up, it starts in real mode and address lines from A20 up are disabled. A block of circuitry called the CPU Reset circuitry produces a signal, which indicates that the microprocessor will only execute instructions in the lower 1 MB of physical memory. In this condition, shadow RAM can be moved into the top of the physical memory space.

The system's A20 line is enabled by writing a logic "1" into the bit-1 position of the keyboard controller's output port. This signal is used to enable the microprocessor's A20 enable pin (AM20). In this condition, the full range of the microprocessor's addressing capability will become available when the microprocessor is reset. At this point, the system performs a special reset operation that affects only the microprocessor. This is accomplished by the software writing a logic "0" into bit-0 of the keyboard controller's output port. This action will reset the microprocessor through the system's reset CPU (RC) line. The microprocessor basically goes to sleep in real mode and wakes up in protected mode.

To switch back from protected mode to real mode, the system must produce an appropriate logic level at the AM20 pin and activate the RC line. This action should not be confused with the system RESET signal, which resets all the devices in the system.

During bootup and initialization, the keyboard controller reads and stores information gathered at its input port. The system's software issues commands to the controller that cause this information to be moved into the output buffer. From this register, the system can check the contents of this buffer for its hardware configuration information. The table also describes the functions of the keyboard controller's input port bits (P1.0–P1.7).

The Output Buffer

The controller also contains an 8-bit, output buffer that duplicates the activities of the SIPO keyboard receiver register, found on PC- and XT-compatible system boards. It receives serial data bits from the keyboard through its T_1 input. These bits are clocked into the keyboard controller's buffer in conjunction with the `KBCLK` signal at the keyboard controller's T_0 input. Under the direction of its internal programming, the keyboard controller shifts the data bits from the keyboard into its output buffer. When the buffer becomes full, the keyboard controller generates the system's keyboard interrupt request (`IRQ1`), and the BIOS's keyboard interrupt handler routine examines the contents of this register to determine which key has been pressed.

On the system board, the `KBCLK` signal is shifted into the controller's T_0 input. Inside the controller, the signal is delayed by two clock periods, and then applied to the output buffer as its clock input. This action synchronizes the data bits to the clock signal to ensure that the data bits are read in the middle of the bit time. The scan code from the keyboard is applied serially to the controller's T_1 input. The two-pulse time delay prevents the scan code's start bits from being shifted into the register. When the eighth bit of the scan code has been shifted into the register, and the keyboard controller receives another clock signal on the `KBCLK` line, it generates and latches the `IRQ1` to the system's interrupt controller.

The interrupt controller responds by placing an `INT9` code on the data bus, which causes the microprocessor to call up the `INT9` keyboard routine from the ROM BIOS program.

The system's BIOS routines convert the scan code into its ASCII-coded equivalent, and stores both codes in a 16×16 FIFO character buffer. This action clears the interrupt request, and allows other interrupts occurring in the system to be serviced. The keyboard buffer is actually organized in the 8-bit address locations of the system's RAM memory. The buffer allows up to 16 characters to be entered before they must be retrieved by the system. The unit beeps when the buffer is full. This arrangement enables the user to enter characters from the keyboard while the system is busy executing a previous command.

The scan codes from the keyboard are stored in the most significant byte of each buffer location; the ASCII-coded character is stored in the corresponding lower byte of each location. Special keys, such as the numeric keypad and function keys (F1 through F12), store a 00 in their lower byte. This differentiates them from the ASCII-coded characters of the keyboard.

To cause the system to read the keyboard entries out of the buffer, the user or the software package must generate an `INT16` signal. This interrupt causes the ROM BIOS routine to retrieve the character codes (both scan and ASCII) from the buffer, and place them in the microprocessor's AX register. The scan code is placed in the `AH` portion of the register; the ASCII code is stored in the `AL` byte. The scan code in `AH` allows the system to differentiate between keys that may return the same ASCII codes. The BIOS routine also checks the status of the codes to see whether any special keys were pressed (Ctrl, Alt, Del, Shift, and so on). If no character has been entered when the interrupt is encountered, and `AH=0`, the `INT16` will wait for a character to be entered.

Finally, the routine sends the ASCII-character code to the program that called for it. The program delivers the code to the activated output device (monitor, modem, or printer). Sending a character to a display device through the CPU is called an `echo`, and may be suppressed by programming so that the character is not displayed.

Troubleshooting Keyboard Problems

Objective

Most of the circuitry associated with the computer's keyboard is contained in the keyboard itself. However, some keyboard interface circuitry is located on the system board. Therefore, the steps required to isolate keyboard problems are usually confined to the keyboard, its connecting cable, and the system board. This makes isolating keyboard problems relatively easy. Just check the keyboard and the system board.

Note

Some of the same symptoms given for keyboard problems are also described in Chapter 9, "Video Display," under the section for video-display problems. This is because most software packages (including DOS) echo keyboard entries, through the system board's RAM memory to the monitor for display.

Keyboard Symptoms

Typical symptoms associated with keyboard failures include the following:

- No characters appear onscreen when entered from the keyboard.
- Some keys work, others don't.
- A `Keyboard Is Locked - Unlock It` error is displayed onscreen.
- A `Keyboard Error - Keyboard Test Failure` error is displayed onscreen.
- A `KB/Interface Error - Keyboard Test Failure` error is displayed onscreen.
- An error code of six short beeps is produced during bootup.
- Wrong characters are displayed onscreen.
- An IBM-compatible 301 error code is displayed onscreen.

Configuration Checks

Keyboard information is stored in the CMOS setup memory, and must accurately reflect the configuration of the system; otherwise, an error will occur. In most CMOS screens, the setup information includes keyboard enabling, Num Lock key condition at startup, `typematic rate`, and `typematic delay`. The typematic information applies to the keyboard's capability to repeat characters when the key is held down. The typematic rate determines how quickly characters will be repeated; the delay time defines the amount of time the key can be held before typematic action occurs. A typical typematic rate setting is six characters per second; the delay is normally set at 250 milliseconds.

As with other components, the only time a configuration problem is likely to occur is when the system is being set up for the first time or when a new option is installed. The other condition that will cause a configuration problem involves the system board's CMOS backup battery. If the battery fails, or has been changed, the contents of the CMOS setup will be lost. After replacing the battery, it is always necessary to run the Setup utility to reconfigure the system.

Software Checks

Turn on the system and observe the BIOS screens as the system boots up. Note the keyboard type listed in the BIOS summary table. If possible, run a selected diagnostic program to test the keyboard. Run the program's Keyboard Tests function, and perform the equivalent of the ALL Tests function if available. These tests are normally very good at testing the keyboard for general operation and sticking keys.

If a stuck key(s) is detected, the individual key switches can be desoldered and replaced with a good key from a manufacturer or a similar keyboard. The amount of time that may be spent repairing a keyboard, however, will quickly drive the cost of the repair beyond the cost of a new unit.

If the keyboard functions properly in DOS but not in Windows, check the Windows keyboard settings. In Windows 3.x, start Windows and double-click on the Main tile. Double-click on the Setup icon, and note the keyboard listed in the Setup window. If the keyboard is not installed, or is incorrect, install the correct keyboard type.

Keyboard Hardware Checks

If a hardware problem is suspected, the first task is to isolate the keyboard as the definite source of the problem. This is a fairly easy task. Because the keyboard is external to the system unit, detachable, and inexpensive, begin by exchanging the keyboard with a known good keyboard. Turn the system on and type characters from the keyboard. If the characters appear correctly on-screen using the new keyboard, check the bottom and back sides of the original keyboard for an 88/286/386 selector switch. If the switch is present, make sure it is set to the correct position for the type of computer being used.

If the selector switch is not there, or if it is in the correct position, return the system to full service and service the defective keyboard appropriately. Remove the back cover from the keyboard and check for the presence of a fuse in the +5 Vdc supply, and check it for continuity. Disconnecting or plugging in a keyboard with this type of fuse while power is on can cause it to fail. If the fuse is present, just replace it with a fuse of the same type and rating.

If the system still won't boot up, recheck the CMOS setup to make sure that the keyboard is enabled. Check the keyboard cabling for continuity. And finally, check the video display system (monitor and adapter card) to make sure that it is functional.

Note

> The process for isolating keyboard problems is reinforced and expanded in the accompanying *Hands-On Lab Book* in procedure 20.

If replacing the keyboard does not correct the problem, and no configuration or software reason is apparent, the next step is to troubleshoot the keyboard receiver section of the system board. In most cases, this involves removing the system unit's outer cover and replacing the system board. After the cover is off, examine the keyboard connector on the system board. Also, look for auxiliary BERG connectors for the keyboard. Make certain that no item is shorting the pins of this connector together. Check for enable/disable jumpers for the keyboard on the system board.

Remove all the options adapter cards from the system board's expansion slots. Disconnect the system board from the power-supply unit (P8–P9) and the system-board/front-panel connections. Take care to mark any connection removed from the system board, and its connection point, to ensure proper reconnection. Exchange the system board for a known good one. Reconnect all the power-supply and front-panel connections to the system board. Reinstall the video and disk-drive controller cards in the expansion slots, and try to reboot the system.

If the system boots up, turn it off, reinstall any options cards removed from the system, replace the system unit's outer cover, and return the unit to full service.

Mouse/Trackballs

Objective

A mouse is a handheld pointing device that produces input data by being moved across a surface. Basically, the mouse is an x-y positioning device that allows the user to move a cursor, or some other screen image, around the display screen. It also enables users to choose options from an onscreen menu—instead of typing in commands from the keyboard or using specialized software—and to create elaborate pictures onscreen. There are currently two methods by which a mouse can produce positional input data.

The first type of mouse is the `optical mouse`. The optical mouse requires a special pad on which it moves. The pad is divided into a number of x and y coordinates by horizontal and vertical lines on the surface of the pad. The mouse detects motion by emitting an infrared light stream that is disturbed when the mouse crosses one of the lines on the pad.

The second mouse type is the `trackball mouse`, depicted in Figure 7.38. This type of mouse detects positional changes through a free wheeling trackball upon which it rides.

Both types of mouse devices have similar appearances, although they may differ in the number of buttons on their top.

In the trackball mouse, movement of the mouse causes the trackball to roll. Inside the mouse, the trackball drives two small wheels that are attached to the rotors of two potentiometers (one x-pot and one y-pot). As the trackball rolls, the wheels turn, and the resistance picked off by the pots' wipers varies proportionally. The varying resistance is converted to a varying analog output voltage. The output voltage undergoes A/D conversion, where it is changed into a digital input that represents movement of the mouse.

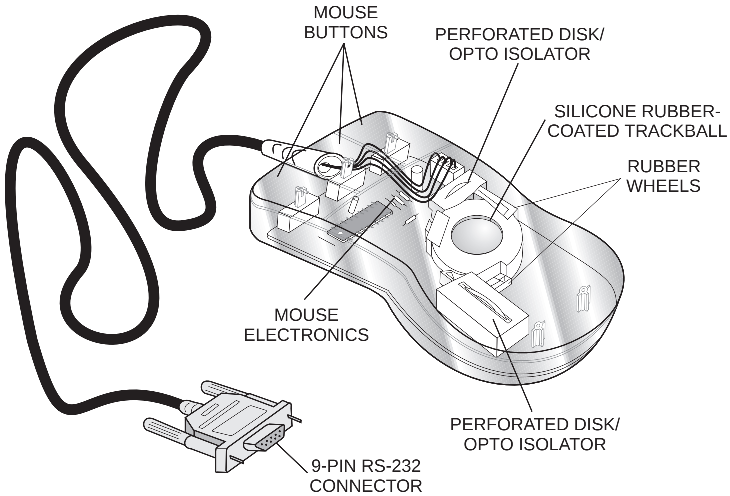


Figure 7.38

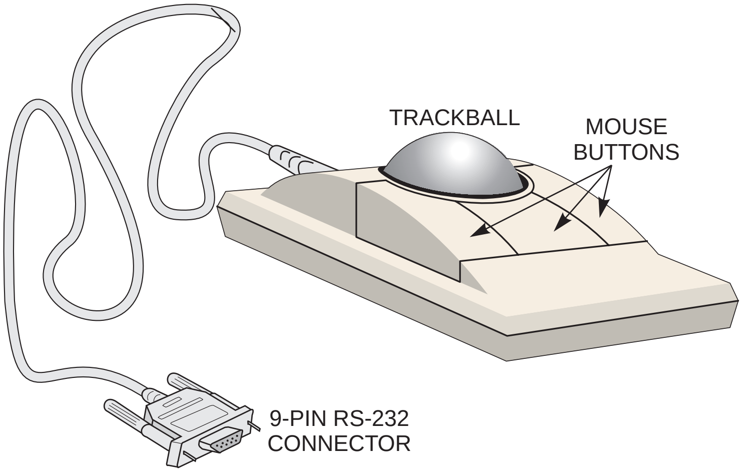
A typical trackball mouse.

Some trackball mice use `opto-coupling` techniques to generate a string of digital pulses when the ball is moved. These devices are referred to as `opto-mechanical` mice. The mouse trackball moves two perforated wheels by friction. Light from LEDs shines through holes in the wheels (which are not attached to potentiometers) as the mouse moves. The light pulses are detected by a photoconductive device, which converts them into digital voltage pulses. The pulses are applied to counters that tabulate the distance (both x and y) that the mouse moves.

In some applications, such as notebook computers, it is desirable to have a pointing device that does not require a surface to be moved across. The trackball can be thought of as an inverted mouse that enables the user to directly manipulate it. Trackballs, like the one depicted in Figure 7.39, may be separate units that sit on a desk or clip to the side of the computer and connect to one of the system's serial ports. In many laptop and notebook computers, trackballs are frequently built directly into the system housing, and connected directly to its I/O circuitry. Like mouse devices, trackballs may come with one to three buttons.

Figure 7.39

A trackball unit.



Mouse Communications

Mice may also be classified by the method they use to communicate with the host computer. The most common type of mouse sends data to the computer `serially`. However, other mice, called `bus mice`, use dedicated cards to connect directly to the system's buses and communicate in parallel with the computer. The serial mouse connects to the system through a DB-9M connector, and uses the RS-232C interface. In older systems, a 25-pin to 9-pin adapter may be required to physically connect the mouse to the unit.

Bus mice use a 9-pin, mini-DIN connector. This prevents them from being erroneously plugged into a serial bus connection. In addition, the bus mouse uses a completely different driver than the typical serial mouse to control its operation.

The circuitry of a typical serial mouse is shown in Figure 7.40. Notice that the typical serial mouse uses only four of the RS-232 serial connection lines. The power for the mouse is obtained from the interface's transmit data (TXD) line and signal ground (GND) line. The TXD line is normally at –12V, and the GND line is at signal ground.

The mouse detects movement through the pulses of light created by the turning, perforated wheels. These wheels rotate between the LED and phototransistor sections of the opto-isolators (ISO1 through ISO4). The intermittent pulses of light cause the transistors to turn off and on in proportion to the distance the trackball rolls. The transistors pulse the x and y inputs of the mouse encoder IC, which determines the speed and direction of movement and generates a pulse train that corresponds to the movement. This information is applied to the serial port through the receive data (RXD) line.

The same type of action occurs when the mouse encoder detects that one of the mouse buttons has been depressed. It generates and transmits a different series of pulses to the interface for each of the various actions of the mouse.

Most mice require special driver software programs to control their operation. The driver software decodes the pulses sent by the mouse, to determine what action the mouse is performing. It also supplies the instructions that interface software applications to the mouse. The driver tracks the position of the mouse based on information from the mouse and your software application. As it receives data from the mouse, it displays a text cursor or graphics pointer onscreen.

Mouse manufacturers normally supply their driver programs with the mouse. In DOS operations, the driver must be installed in the system before using the mouse. If the driver file is included in the AUTOEXEC.BAT file, it will be loaded into memory each time the computer is booted. Conversely, some mouse drivers are loaded in CONFIG.SYS as device drivers. As mentioned in Chapter 3, the installation of the driver software may be performed automatically by the installation software, or need to be done manually by the technician/user. This depends on the nature of the device manufacturer's installation program.

Typically, a MOUSE.COM file is called for in the AUTOEXEC.BAT file, and a MOUSE.SYS file will be referenced in the CONFIG.SYS file. Driver software for several common mice are included in the Windows software package. These drivers can be installed and set up at the time Windows is installed. The driver setup can also be changed anytime a new mouse is installed, by accessing the Setup window.

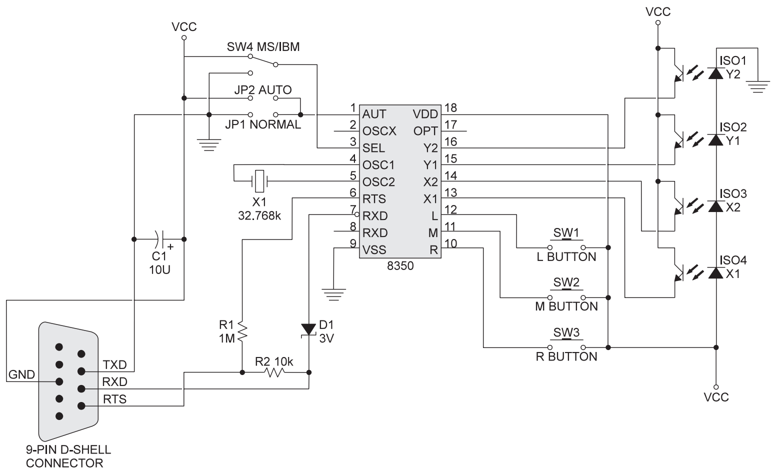


Figure 7.40

Mouse circuitry.

Windows Mouse/ Trackball Operation

The Windows software provides drivers for a number of different mice. The action of the mouse can be changed through the Mouse option of the Control Panel. To do this, select the Control Panel icon from the Main window. Double-click on the Mouse option and move into the Mouse dialog box, depicted in Figure 7.41.

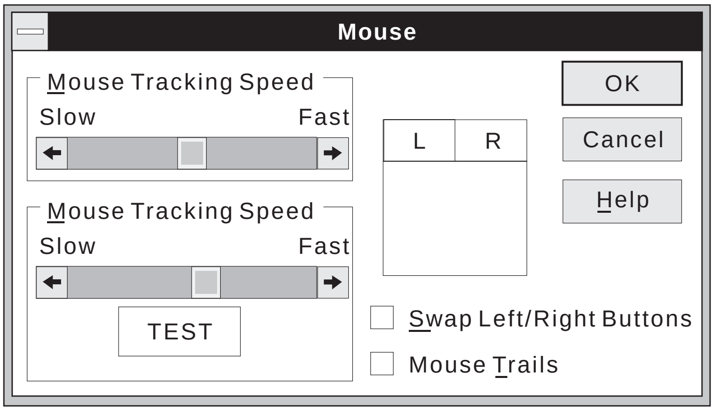


Figure 7.41

The Windows 3.x Mouse dialog box.

The mouse tracking speed and double-click speed can be modified from this dialog box. The tracking speed determines how fast the mouse pointer moves onscreen when you move the mouse. This option can be varied between the extremes of slow and fast. The tracking speed is changed by clicking the mouse on the arrow beside each extreme, to move the scroll bar in that direction.

The double-click speed is changed in the same manner. Often, the speed at which the user double-clicks the mouse button increases with more use of Windows. This can lead to frustration when you try to activate an icon and double-click too fast. After a new setting is made, the double-click speed can be sampled through a test box under the option. The test box is highlighted when a valid double-click is recognized, according to the current setting of the double-click speed option.

The Mouse dialog box also enables the user to change the function of the mouse buttons for those who are left-handed. In the bottom right-hand corner of the dialog box is a Swap Left/Right Buttons check box. By clicking on this check box, an x is placed in the box to indicate that the buttons are now reversed. Under this check box is another check box labeled Mouse Trails. When activated, it causes mouse movement to leave a trail of mouse pointers onscreen.

Troubleshooting Mouse Problems

The levels of mouse troubleshooting move from configuration problems to software problems—including DOS, Windows, and applications—to hardware problems.

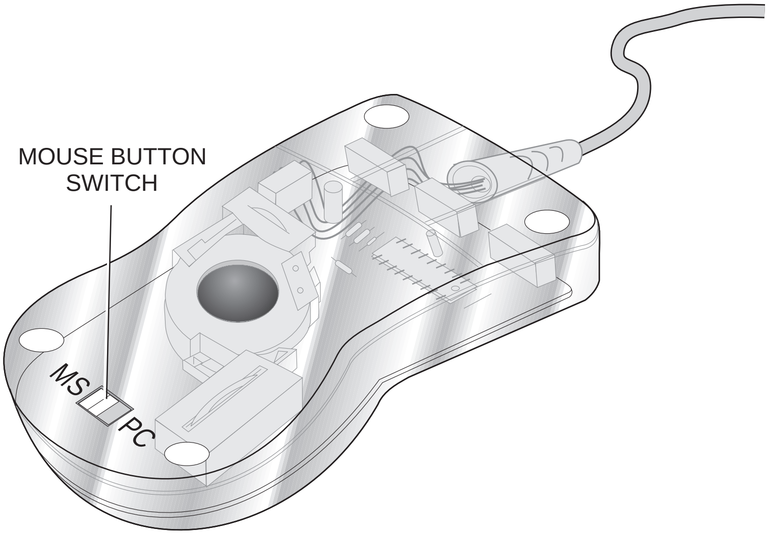
Maintenance of the mouse is fairly simple. Most of the problems with mice relate to the trackball. As the mouse is moved across the tabletop, the trackball picks up dirt or lint, which can hinder the movement of the trackball. This is typically evident by the cursor periodically freezing and jumping on the screen. On most mice, the trackball can be removed from the mouse by a latching mechanism on its bottom. Twisting the latch counterclockwise allows the removal of the trackball. Dirt can then be cleaned out of the mouse.

Mouse Configuration Checks

A common problem encountered when a new mouse is being installed is the setting of its 2/3 button switch. This switch, depicted in Figure 7.42, is located on the bottom of the mouse and may be marked with a 2/3 designation, or as PC/MS. The PC designation stands for 3-button operation; the MS (Microsoft) setting specifies 2-button operation. This switch setting must correspond to the software driver configuration.

Figure 7.42

The 2/3 button selection switch.



Mouse Software Checks

Test the operation of the mouse in DOS first. Most mouse drivers come with a built-in test for DOS operation. If so, run the test program from the Mouse directory. If no self-test software is included, check the mouse in a non-Windows application.

If the mouse does not operate in DOS, check the BIOS setup screen during bootup for the presence of the serial port that the mouse is connected to. Check the directory structure of the system for a mouse directory. Examine the AUTOEXEC.BAT and CONFIG.SYS files to see where the system looks for the mouse drivers. Two common driver files may be present. These are the MOUSE.COM file called for in the AUTOEXEC.BAT file, and the MOUSE.SYS file referenced in the CONFIG.SYS file.

If the driver software is not present in the directory indicated by the AUTOEXEC.BAT file, it will need to be installed. At the C:\>> DOS prompt, type **MD\MOUSE**. Then, type **CD\MOUSE** to move into the new subdirectory. Locate the manufacturer's driver disk and insert it into A: drive. Copy the mouse driver software into the MOUSE subdirectory on C: drive.

If the mouse functions properly in DOS but not in Windows 3.x, the Windows mouse driver and mouse settings need to be checked. Begin by checking the Windows mouse driver in the Setup window. Move into the Main window by clicking on its tile, and then click on the Setup icon. Click on the Options entry from the toolbar, and choose the Change System Settings option from the menu. Use the scroll arrow at the right of the window to move through the available driver options. Select the correct driver from the list, if available.

If the correct driver is not available in the Windows list, place the manufacturer's driver disk in the floppy drive, and load it using the Other Mouse (requires disk from OEM) option. If the OEM driver fails to operate the mouse in Windows, contact the mouse manufacturer for an updated Windows driver. Windows will normally only support mice on COM1 and COM2. Therefore, check the COM settings under the Control Panel's Ports icon. If several serial devices are being used in the system, it may be necessary to establish alternative IRQ settings for COM3 and COM4.

Check the IRQ and base address settings for Windows. Compare these settings to the actual configuration settings of the hardware. If they differ, change the IRQ or base address setting in Windows to match those of the installed hardware. Check the AUTOEXEC.BAT and CONFIG.SYS files for conflicting device drivers. In particular, look for a `DEVICE =` command associated with the mouse. Also check for `Device` statements in the CONFIG.SYS file that apply to the mouse.

In Windows 95, click on the Mouse icon in the Control Panel window to check the configuration and settings for the mouse. Follow this by checking the port configuration in the Control Panel. Consult the Device Manager entry under the Control Panel, System icon. Select the Ports option, click on the COM*x* Properties option in the menu, and click on Resources. Make certain that the selected IRQ and address range match that of the port.

Mouse Hardware Checks

> The process for isolating mouse problems is reinforced and expanded in the accompanying *Hands-On Lab Book* in procedure 21.

If the 2/3 button switch and driver setup are correct, it will be necessary to divide the port circuitry in half. For most systems, this involves isolating the mouse from the serial port. Just replace the mouse to test its electronics.

If the replacement mouse works, the original mouse is probably defective. If the electronics are not working properly, there are few options available for servicing the mouse. It may need a cleaning or a new track ball. However, the low cost of a typical mouse generally makes it a throwaway item if simple cleaning will not fix it.

If the new mouse does not work either, chances are very high that the original mouse's electronics are working properly. In this case, the driver software or port hardware must be the cause of the problem.

Joysticks and Game Paddles

Objective

Joysticks are very popular peripheral devices for use with computer video games. These devices do, however, also provide a convenient computer/human interface for a number of other applications. Nearly every computer in the personal/business category includes ports for joysticks.

Most joysticks are x-y positioning devices with a handle (gimbal) that can be moved forward, backward, left, right, or at any angular combination between these four basic directions. The joystick converts the movement of the handle, along either axis, into a digital signal that can be used to move a cursor on a display or that can be processed as data input.

There are two major types of joysticks, `analog` and `digital`. The analog version illustrated in Figure 7.43 employs two resistive potentiometer elements, one for x-direction and one for y-direction. Both potentiometers are mechanically connected to the movable gimbal. A DC voltage applied across the resistance elements produces variable levels of output voltage when the gimbal is moved along the x-axis, y-axis, or at some angle between them (this varies both the x and y voltages). These analog voltages are applied to A/D converter circuits, which produce digital x-y coordinate information for input to the computer. When this type of joystick is used to position a screen image, the position of the image on-screen corresponds to the x-y position of the gimbal.

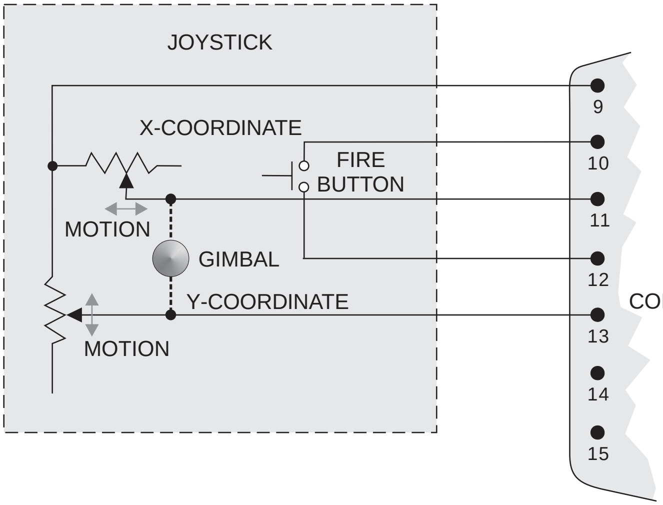


Figure 7.43

An analog joystick.

Figure 7.44 shows a simplified version of the A/D conversion process used with joysticks. The portion of the joystick's resistance picked off by the position of the gimbal becomes part of an RC proportional time constant that controls the timing factor of a one-shot timer. The duration of the one-shot's output pulse (Tp) is determined by the value of the RC time constant. More resistance produces a longer pulse duration; less resistance causes the timing pulse to be shorter.

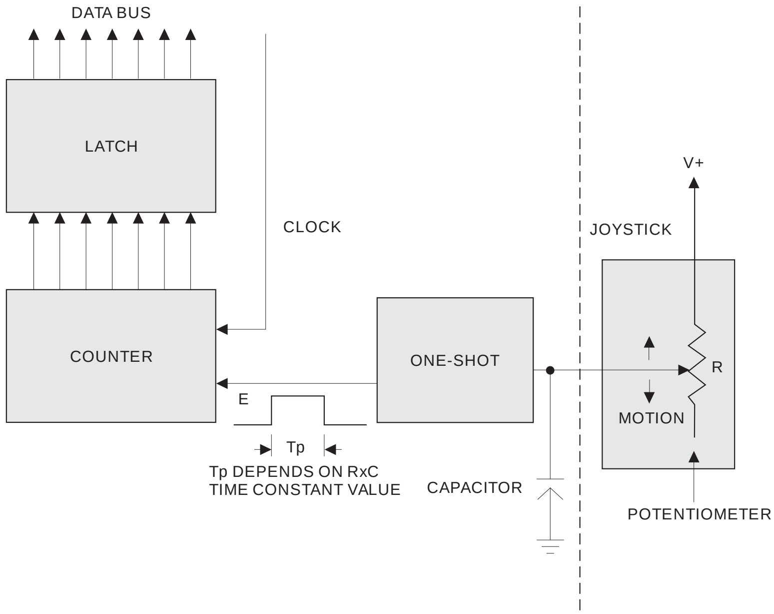


Figure 7.44

A-to-D conversion.

The output of the timer is used to clear and enable a digital pulse counter. It counts pulses produced by an oscillator having a predetermined output frequency. If the counter is enabled for a relatively long duration, the binary count will be higher than if the counter were enabled for a shorter time period. At the end of the enabling pulse, the output of the counter is latched into a PIPO register, where the system can read it. The counter is then cleared; after a short reset delay, the one-shot is re-triggered for another count sequence.

A somewhat simpler design is used in the construction of digital joysticks. The gimbal is used to mechanically open and close different combinations of an internal, four-switch arrangement, as depicted in Figure 7.45. The joystick produces a single-byte output, which encodes the gimbal's movement in any of eight possible directions. Unlike analog joysticks, the position of the controlled image onscreen does not correspond to the x-y position of the gimbal. Instead, the gimbal position only produces the direction of movement for the screen image. When the gimbal is returned to its neutral position, the screen image just stops where it is.

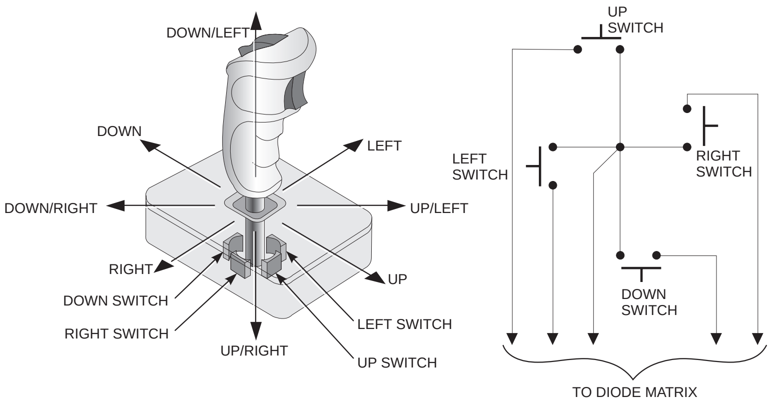


Figure 7.45

A digital joystick.

Game Port

The game port allows game paddles or joysticks to be used with the system. The port's interface circuitry converts resistive input values into relative paddle or joystick positions in much the same manner as described in the preceding section. This port can also function as a general-purpose I/O converter, featuring four analog and four digital input points.

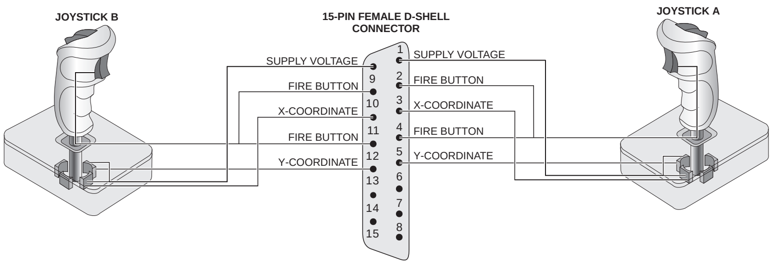
The input to the game port is a set of resistive joysticks or game paddles. Joysticks are defined as having two variable resistances, and game paddles as consisting of one variable resistance. Each resistance should be variable between 0 and 100 k ohms. Joysticks may have one or two normally open (fire) buttons. Game paddles have only one button. Under this definition, the games adapter will support two game paddles (A and B), four game paddles (A, B, C, and D), or two joysticks (A and B). The order of fire buttons should correspond to that of the resistive elements (A and B, or A, B, C, and D).

This interface is simple and straightforward, combining elementary hardware and software techniques. When the software issues the adapter's address (201_h) and an `IOW active` I/O write signal, the timers are triggered into their active state, and their outputs go to a high-logic level. The individual timer outputs remain high for a length of time, dictated by the setting of each resistive input and timing capacitor at each timer's input. As each timer times out according to its own RC time constant, its output returns to a low-logic state.

Figure 7.46 shows a connection scheme for attaching joysticks to the 15-pin, D-type connector at the rear of the system. Unlike the 3-row, 15-pin D-shell connector used for VGA video, the game port connector has only two rows of pins. When attaching game paddles to the adapter, four separate +5 Vdc supplies are furnished (pins 1, 8, 9, and 15), one for each paddle. The adapter also provides separate grounds for paddles A and B (pins 4 and 12). Paddles C and D may also use these grounds, or they may share a ground connection at pin 5. The wipers of the resistance elements are connected to pins 3, 6, 11, and 13 (A, B, C, and D). The order of the fire buttons should correspond to that of the resistive elements. Therefore, connect the A, B, C, and D buttons to pins 2, 7, 10, and 14, respectively. The buttons should be of the normally open variety.

Figure 7.46

Joystick connection.



Game Software

The game software periodically polls the adapter's data bus buffer/driver latch to determine whether each output has timed out. A software counter keeps track of the number of times the port has been read before each timer times out. The number of read cycles (0 to 255) before a logic low is encountered is directly proportional to the resistive setting of the joystick or game paddle.

The fire buttons are read by the software in the same manner as the resistive inputs. In the button's inactive (open) state, the digital inputs are at a high-logic state. When a button is depressed, its input is pulled to a low-logic level. When the software reads the output latch, it detects a low at the bit position reserved for that button and interprets it as a closure.

Troubleshooting Joystick Problems

As with other input devices, there are three levels of possible problems with a joystick. These are configuration, software, and hardware. Attempt to run the joystick with a DOS-based program. If the joystick will not work at the DOS level, check the game port's hardware configuration settings. Compare the hardware settings to that of any software using the game port. Try to swap the suspect joystick for a known good one.

Windows 3.x makes no provision for a joystick device. However, Windows 95 contains a joystick icon in the Start/Settings/Control Panel window. Figure 7.47 shows the contents of the Win95 Joystick Properties dialog box. The Joystick icon is not loaded into the Control Panel window if the system does not detect a game port during installation. If the port is detected during bootup, but no joystick device is found, the window will say that the joystick is not connected correctly.

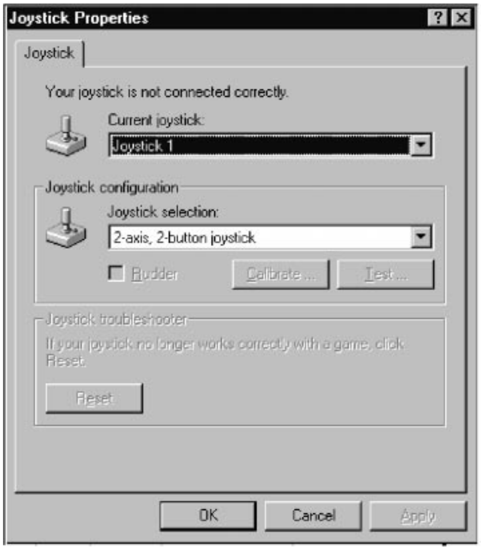


Figure 7.47

The Joystick Properties dialog box.

The Joystick Properties dialog box is used to select different numbers of joysticks (1–16) that can be used. It shows the currently selected joystick type, and allows other devices to be selected. The default joystick setting is a 2-axis, 2-button device. You can also select None or Custom joystick types. Other selections include 2-axis, 4-button joysticks and 2-button game pads, as well as specialized flight yoke and flight stick assemblies.

The Joystick Properties dialog box also contains a button to calibrate the joystick's position. This button allows the zero-position of the stick to be set for the center of the screen. The Test button allows the movement of the stick and its button operation to be checked, as directed by the test program. Win95 also allows for a rudder device to be added to the system. The Joystick Troubleshooter section contains a Reset button to re-initialize the game port. This function is normally used if the stick stops responding to the program.

Light Pens

Objective

Light pens, as depicted in Figure 7.48, are handheld light-sensing devices that work in conjunction with the scanning electron beam of a CRT display. As the electron beam sweeps the screen of the CRT, it "paints a picture" on the face of the screen by selectively striking phosphor spots located on its face. When the electron beam strikes a phosphor, the phosphor glows, leaving a spot of light that remains for a predetermined length of time. The length of time depends on the type of phosphor and the intensity of the electron beam when the phosphor was struck. The location of each phosphor spot has a specific x and y coordinate on the face of the screen. The light pen uses the scanning electron beam to produce x/y-coordinate information about the position of images onscreen for input into the CRT controller.

Light Pen Operation

The light pen contains a light-sensitive device, such as a phototransistor or photo-diode, which produces a pulse when struck by the passing CRT electron beam. The monitor's CRT controller can determine the exact position of the light pen against the face of the screen by checking the horizontal and vertical position of the scanning electron beam at the time it receives the pulse from the light pen.

Figure 7.48

A typical light pen.

The CRT controller obtains the y-coordinate (number of raster scans down the screen) and the x-coordinate (number of dot clock pulses since the last horizontal synchronization [HSYNC] signal) from the horizontal and vertical counters used to access the CRT's screen memory. These values are combined, and loaded into the CRTC's light pen address register, when enabled by the pulse from the light pen. At the same instant, the CRTC sends an interrupt input signal to the computer.

The light pen may seem like a very simple device, and electronically it is. Some simple light pen designs are available through hobby-magazine articles. But a truly useful and accurate light pen must overcome some very troublesome design problems (see Figure 7.49). In addition to the fact that the space allotted for electronics in the light pen is somewhat limited, the light pen must be capable of resolving the electron beam's trace accurately (resolution), and reject noise and unintended light from being interpreted as input (ambient light rejection).

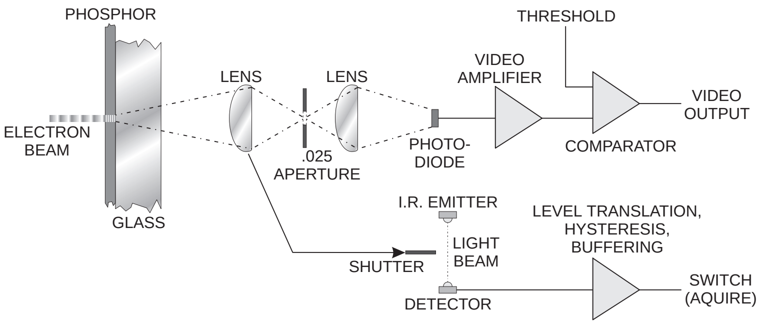


Figure 7.49

Inside the light pen.

The pen uses a mechanical aperture, which is opened when the pen is pressed against the screen. An optical lens is used to restrict the area of the screen that the pen "sees." Figure 7.50 shows how the aperture and lens arrangement prevents the photosensitive device from detecting false traces of the electron beam.

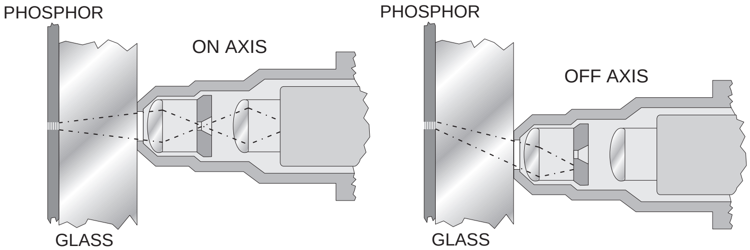


Figure 7.50

Light pen focusing.

When a phosphor strike occurs within the light pen's field of focus, the photo detector's characteristics change and a digital pulse is created. The pulse is amplified and applied to a level comparator, which compares the amplitude of the pulse to some predetermined threshold level. If the input pulse is large enough, the comparator passes the pulse to the CRT controller as a valid strike.

Light Pen Parameters

A number of factors must be taken into account when setting the threshold level of the pen. Primary factors of concern are the color and type of phosphor used in the monitor and the intensity level at which the screen is set. Phosphor characteristics, such as persistence and peak light-emission amplitude, can be problematic in that long-persistence phosphors such as P39s can cause false strikes to occur. Likewise, low-peak amplitude phosphors (especially red) may not produce enough amplitude to register a strike.

Phosphor color and type can be a problem in monochrome monitors as well as color monitors. Monochrome monitors use a single color phosphor, but that color varies from manufacturer to manufacturer. This point should not be overlooked when considering a light pen for use with a particular monitor.

On the other hand, the intensity (brightness) setting of the monitor itself may produce similar problems by being set too low or too high. Other factors that affect the response of the light pen are the response characteristics of the photodetector itself, the type and thickness of the glass face plate, and the distance the photodetector is actually removed from the screen. All these factors can cause delays that can produce false strikes, because the CRT controller counters have advanced by the time the strike is noticed. The fixed delays can be overcome, however, by proper software considerations.

Depending on the software being used, the location specified by the light pen can be used to address the screen memory so that the computer can examine the contents of that location. The computer can then use this information to identify the character, or image of interest, onscreen.

The attribute most commonly associated with light pens is that they provide an easy means of choosing commands, options, or programs from an onscreen menu. The light pen also provides the naturalistic method of drawing or creating onscreen graphs and diagrams. Software packages for such light pen applications usually contain routines to draw circles, boxes, or lines, to enlarge or reduce the size of graphic images, and to shade or change the color of different areas of the screen. More advanced software may enable the user to create truly artistic renderings onscreen.

Troubleshooting Light Pen Problems

There are basically four components to consider when troubleshooting light-pen–related problems. These are the light pen, the monitor, the video adapter card, and the light-pen software.

Depress the pen's aperture and listen carefully trying to hear the mechanical click of the shutter. Examine the light pen's connection to the system and to the video adapter card to see that it is correct and making good contact at each point.

Check the monitor's brightness and intensity settings to see whether the pen functions at different settings of each.

Review the light pen's user's manual and software setup for possible configuration problems and IRQ conflicts. Also, review the monitor's specifications and compare them to those of the light pen to make sure that they are compatible.

Reinstall the light pen software, carefully reviewing each step. Check for the presence of diagnostic routines in the light pen software. Check the video adapter by substitution. Check the light pen by substitution.

Touch-Sensitive Screens

Hewlett-Packard introduced the first touch screen monitor in 1983. Touch-sensitive screens divide the screen into rows and columns that correspond to x and y coordinates. There are two common methods for accomplishing this. The first method uses see-through membranes arranged in rows and columns over the screen, as illustrated in Figure 7.51.

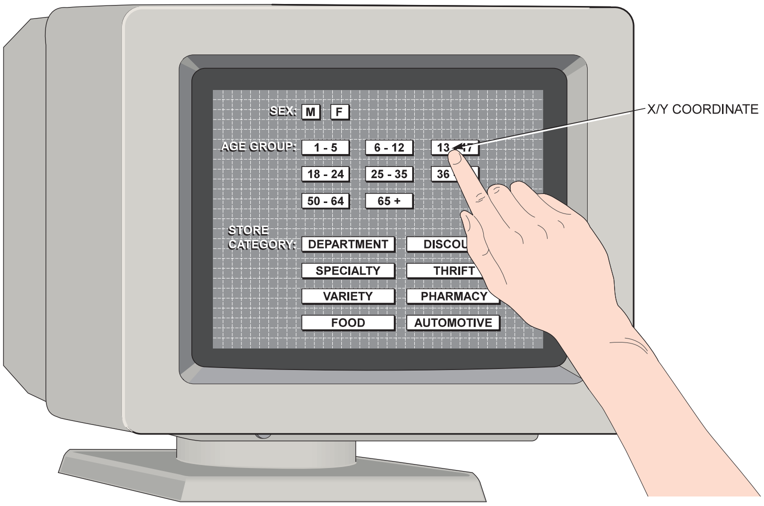


Figure 7.51

Crossing strips.

When the user presses the touch-sensitive panel, the mylar strips are pressed together. When strips from a row and a column make contact with each other, their electrical qualities change. The signal generated between the two strips is decoded to an approximate x/y position onscreen by the panel's decoding circuitry.

The second type of touch-sensitive screen technology employs infrared techniques to section the screen. Banks of LEDs and sensors arranged around the face of the monitor, as illustrated in Figure 7.52, divide the screen into a grid pattern. When an object interrupts the signal paths between a pair of horizontal and vertical LEDs and sensors, a decodable signal is produced that can be related to an x/y coordinate onscreen.

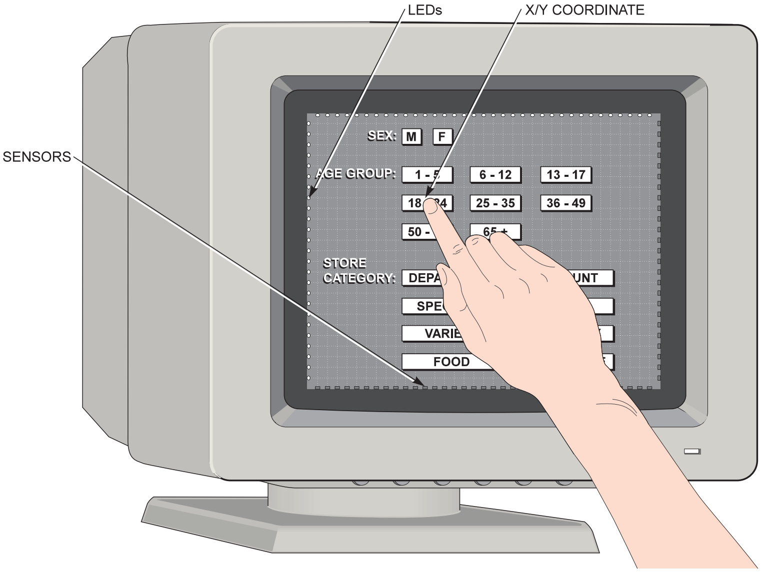


Figure 7.52

An LED sensor-detection arrangement.

The main drawback associated with using touch screens involves the excessive arm movements required to operate the system. It is also true that the human fingertip is not a fine-enough pointing device to select small points onscreen. Therefore, the location of a small item pointed to onscreen may not be exact due to the relative size of the fingertip. The software designer must create screen displays that take this possibility into account, and compensate for it where touch screens are used.

Touch-sensitive panels are available as built-in units in some monitors; other units are designed as add-ons to existing monitors. These units clip or strap to the body of the monitor, and hang down in front of the screen. In add-on units, the coordinate mismatch problem can be compounded by the addition of `parallax errors`. These errors are caused by the distance between the screen and the sensors, and the angle at which the user views the display. Parallax error causes the image to appear at a different location than it actually is. This concept is illustrated in Figure 7.53.

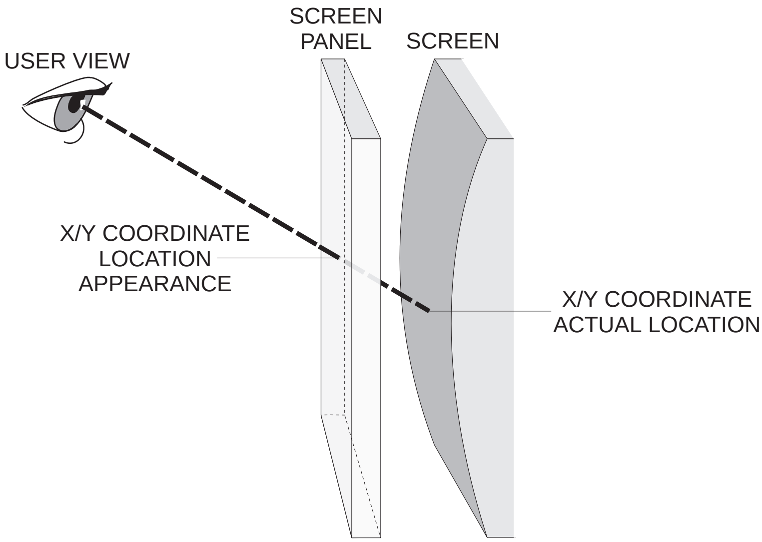


Figure 7.53

Parallax error.

Troubleshooting Touch-Sensitive Screen Problems

There are really only three components to consider when troubleshooting problems associated with touch-sensitive screens. These are the touch-sensitive screen (or the monitor it is used with), the I/O port it is attached to, and the screen's driver software.

Review the screen's user's manual and software setup for possible configuration problems. Examine the I/O port connection and configuration to make sure it is properly set up to support the touch screen. If the screen is an add-on unit, check the monitor's specifications to make sure that they are compatible with the touch-screen unit.

Reinstall the touch-screen software, carefully reviewing each step. Check for the presence of diagnostic routines in the screen's software. Check the I/O port settings. Check the touch-screen unit by substitution.

Scanners

Objective

Scanners convert pictures, line art, and photographs into electronic signals that can be processed by software packages such as desktop publishers and graphic-design programs. These programs, in turn, can display the image on the video display or print it out on a graphics printer.

The scanner borrows technology from the office copy machine. An image is scanned with a light source, and the level of light reflected from the image is applied to a light-sensitive receiver. Unlike the copier, the image is not left on the light-sensitive receiver. Instead, it is converted into an electronic signal stream that can be manipulated by the computer.

Scanners basically come in two types: `handheld scanners`, and `flat-bed scanners`. Handheld scanners tend to be less expensive than flat-bed scanners due to less complex mechanics. However, handheld scanners also tend to produce lower-quality images than flat-bed scanners. They normally require two passes to scan an entire page-sized image; flat-bed scanners can pick up the complete image in one pass. The handheld scanner can be used to scan images from large documents or from irregular surfaces. The quality and accuracy, however, depend on the steadiness of the user.

Scanners can also be classified by the type of images they can reproduce. Some scanners can only differentiate between different levels of light and dark. These scanners are called `grayscale scanners`. `Color scanners`, on the other hand, include additional hardware that help them to distinguish between different colors.

Handheld Scanners

The handheld scanner depicted in Figure 7.54 operates by being pulled across an image. The user begins the scanning process by pressing the Scan button, and then moving the scanner body across the image. An LED in the scanner projects light on the image as the scanner moves. As the light passes over darker and lighter areas of the page, varying levels of light are reflected back to a focusing lens.

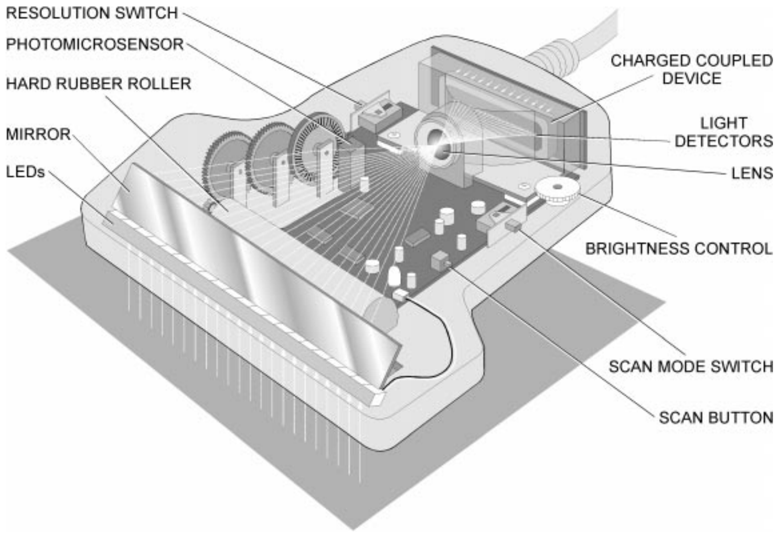


Figure 7.54

Inside a handheld scanner.

The lens focuses the reflected light stream onto a `charge-coupled device` (CCD), which converts the intensity of the light into a proportional voltage signal. The CCD is the same type of device used in the lens of a typical camcorder. The voltage level produced corresponds to black, gray, and white light levels.

Internal circuitry processes the voltage signal before passing it to a special adapter card in one of the computer's expansion slots. The scanner's encoding circuitry performs `gamma correction` on the signal to enhance the black levels of the image. This process alters the image to make it more appealing to the human eye. This is necessary because of the eye's tendency to be more sensitive to darker tones than lighter ones.

The color of the light source used in the scanner also affects how the human eye will perceive the output. When scans of color material are taken, the color of the scanning light used may not produce brightness levels compatible with how the human eye perceives it. This is true in both color and grayscale output. The two most common scanner light source colors are red and green. The green light produces output that looks much closer to the way the eye perceives it than the red light can. For line art and text scans, the color of the light source is not important.

After gamma correction, the voltage signal is applied to an `analog-to-digital converter` (ADC). The ADC samples the voltage signal present at its input and generates a corresponding digital value. In low-level handheld scanners, the digital output is an 8-bit value that can represent up to 256 different shades of gray.

As the scanner moves across the surface, a wide rubber roller turns a series of gears, which in turn rotate a perforated disk. An `opto isolator` shines a light through the slots in the disk as it turns. The light strikes an optical sensor on the other side of the disk and produces pulses as the spokes of the disk interrupt the light stream. The pulses created by the disk are used to coordinate the transmission of the digitized image values with the movement of the scanner. Each time a line of image data is transmitted to the adapter card, the ADC is cleared out and begins gathering a new line of image data. Figure 7.55 illustrates how the scanner's mechanical parts are used to coordinate the image gathering and transmitting process in a handheld scanner.

The software included with most scanners provides the user with at least a limited ability to manipulate the image after it has been scanned. Because of the limited width of most handheld scanners, the software also provides for integrating two consecutive scans to form a complete picture.

Older handheld scanners provided scanning resolutions up to 300 `dots per inch` (dpi). Common scanning resolutions are 600 and 1200 dpi. Newer, color handheld scanners can produce 24-bit, high-resolution (3200 dpi) image editing with 16 million colors.

Flat-Bed Scanners

Flat-bed scanners differ from handheld units in a couple of areas. First, in a flat-bed scanner, the scanner body remains stationary while a scan head moves past the paper. This process is illustrated in Figure 7.56. The paper is placed face down on the scanner's glass window. The light source from the scanning mechanism is projected up through the glass and onto the paper. The lighter areas of the page reflect more light than the darker areas do.

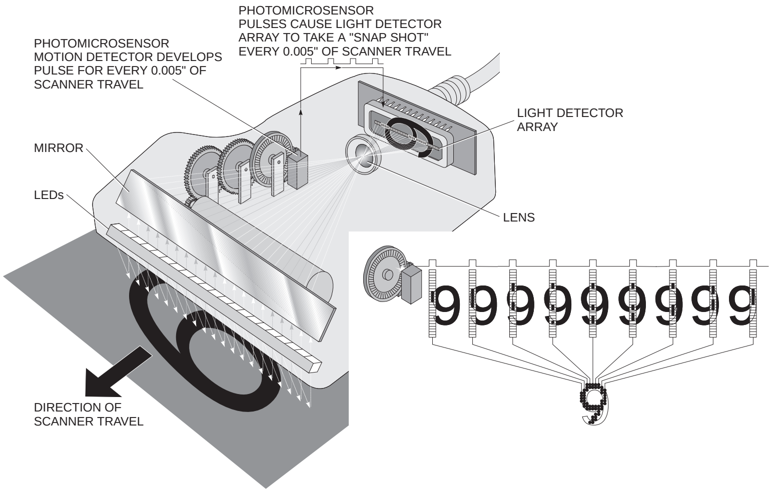


Figure 7.55

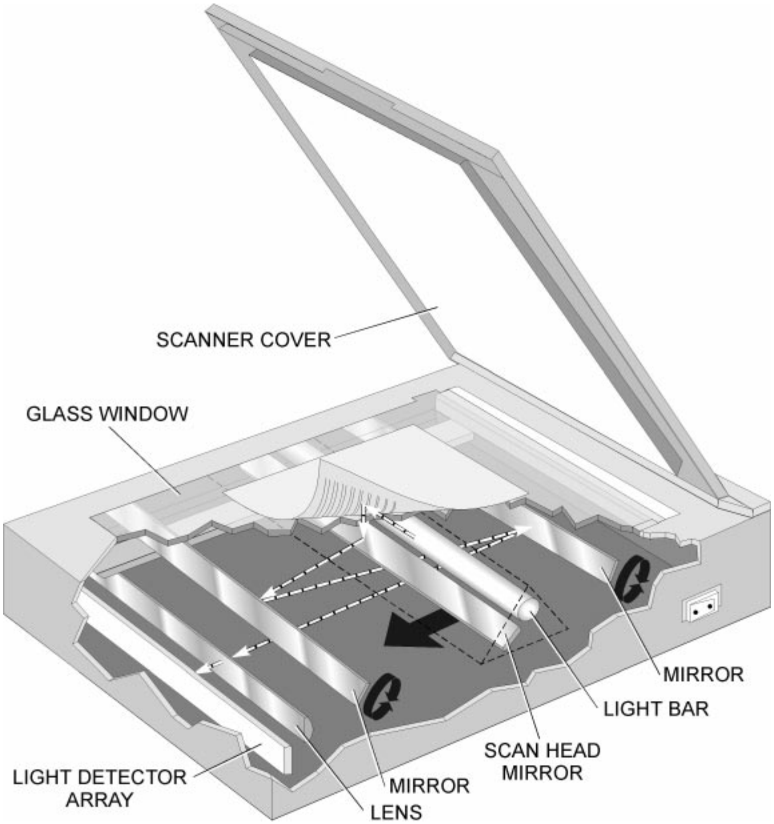
The scanner's mechanical structure.

A precision positioning motor moves the scan head below the paper. As the head moves, light reflected from the paper is captured and channeled through a series of mirrors. The mirrors pivot to continually focus the reflected light on a light-sensitive diode. The diode converts the reflected light intensity into corresponding electrical voltages that are applied to an analog-to-digital converter.

The ADC converts the analog voltages into corresponding digital values through the A-to-D process. Each digital value corresponds to a pixel of page information. A normal scanner resolution is 300 dots (or pixels) per inch. Newer flat-bed scanners can achieve resolutions up to 4800 dpi. At these resolutions, each dot corresponds to about 1/90,000 of an inch. The higher the selected scanning resolution, the slower the computer and printer will run, because of the increased amount of data that must be processed.

Figure 7.56

A flat-bed scanner.

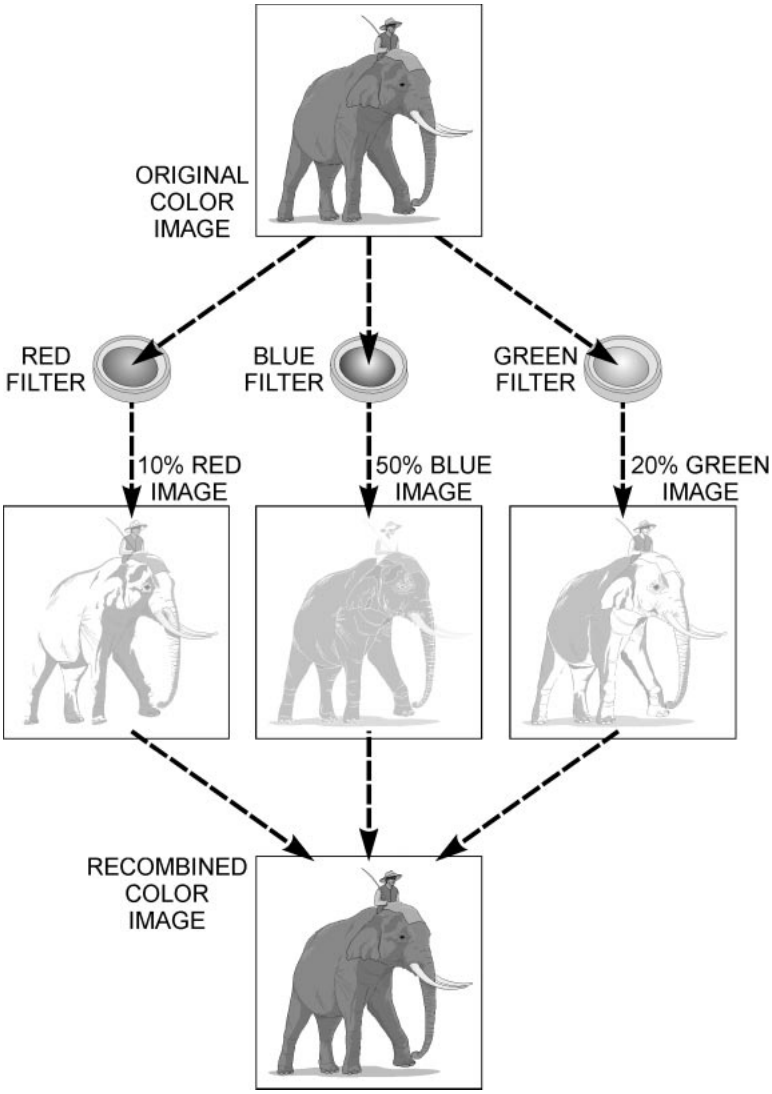


The digitized information is routed to the scanner adapter card in one of the PC's expansion slots. In main memory, the graphic information is stored in a format that can be manipulated by graphic-design software.

Grayscale scanners can differentiate between varying levels of gray on the page. This capability is stated in `shades`. A good-quality grayscale scanner can differentiate between 256 levels of gray. Color scanners, on the other hand, use three passes to scan an image. Each scan passes the light through a different color filter to separate them from each other. The red, blue, and green filters create three different electronic images that can be integrated to form a complete picture. For intermediate colors, varying levels of the three colors are blended to create the desired shade. This concept is illustrated in Figure 7.57.

Figure 7.57

Color filters.



Like the handheld scanners, most flat-bed scanners use a proprietary adapter card and cable to communicate with the system. A number of SCSI-interfaced scanners are available, however. One of the most common problems with installing scanners involves finding a vacant expansion slot for the adapter card. This is particularly true with Pentium boards that use a mixture of ISA and local bus slots. To overcome this problem, scanners are now being produced that operate through the system's parallel printer port.

In these units, the printer plugs into the scanner, which, in turn, connects to the port. In older units, such as PCs and XTs, the limited number of available interrupt request lines often became a problem.

Both types of scanners require an adapter card. Although most use proprietary adapters, SCSI adapters are also widely used with scanners.

Optical Character Recognition (OCR)

Objective

Scanners are also used to scan written text from a page and convert it into characters that can be manipulated by the computer. In these situations, special `optical character recognition` (OCR) software is used to recognize written characters and convert them into computer characters. In some types of operations, the capability to convert nonelectronic text into an electronic format usable by the computer is a great time and effort saver.

When the scanner passes over a character, areas of light and dark are sensed. The OCR software maps these areas into a character grid pattern similar to the one in Figure 7.58.

The software then compares the scanned character to character patterns it has stored in its library, until a near match is located.

One of the biggest problems with OCR systems lies in their limited capability to recognize different font versions of a character. In addition, the OCR system must be able to overcome spots, stains, colored paper, ink colors, and creases in the paper. Misalignments of the paper being scanned can also cause errors to occur in the character-pattern–recognition process.

Troubleshooting Scanners

The driver software that comes with the scanner must be configured to match the settings on its adapter card. The adapter card settings are normally established through hardware jumpers. Check the scanner software's setup screen to confirm the settings.

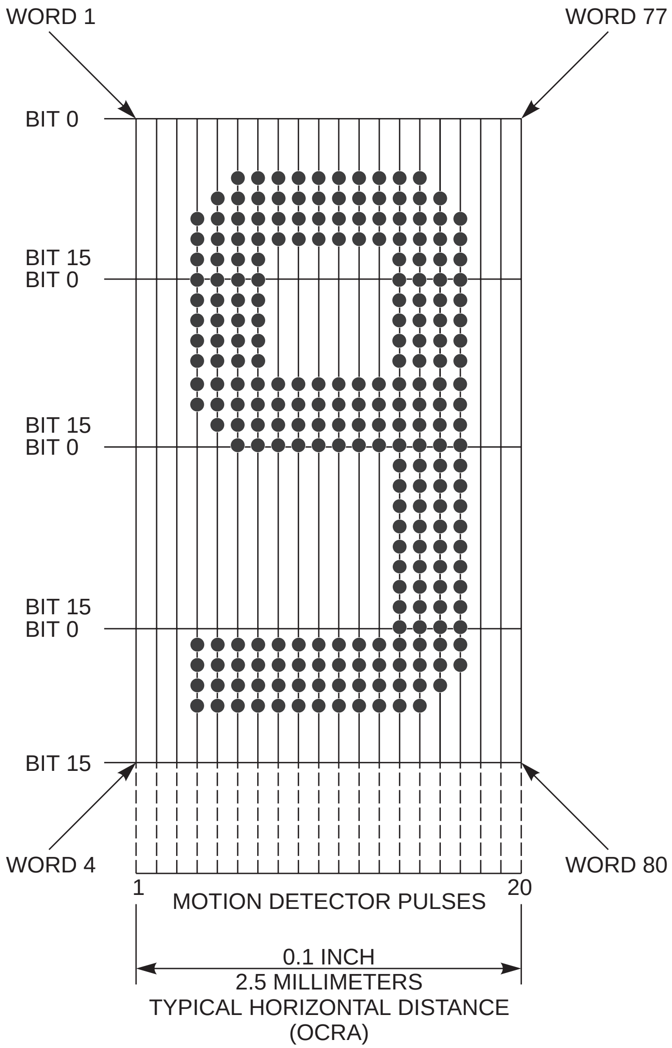


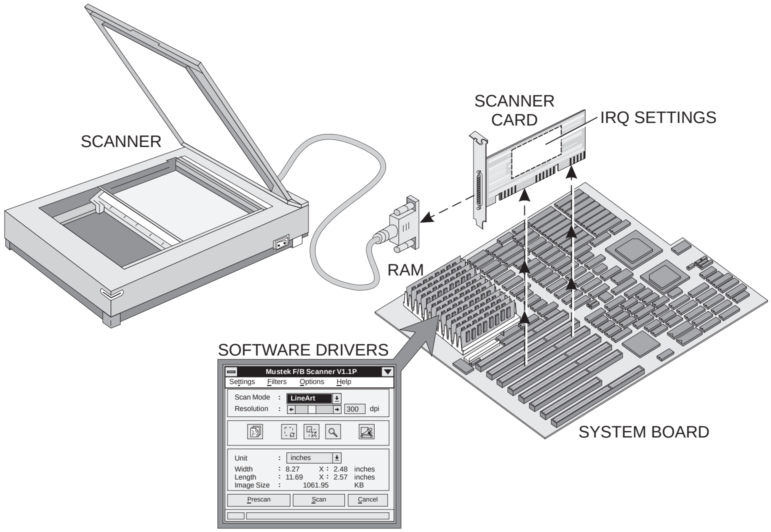
Figure 7.58

Optical character recognition.

Most scanners have three important configuration parameters to consider. These are the I/O address, the IRQ setting, and the DMA channel setting. Typical I/O address settings for the scanner adapter are 150_h to 151_h (default), 170 to 171, 350 to 351, or 370 to 371. Typical IRQ settings are 10 (default) 3, 5, and 11. Likewise, typical DMA channel settings are channel 5 (default), 1, 3, or 7. Figure 7.59 shows the components associated with scanners.

Figure 7.59

Scanner-related components.



Traditionally, IRQ conflicts with network and sound cards tend to be the biggest problem associated with scanners. Typical symptoms associated with IRQ conflicts include the following:

The image onscreen appears misaligned.

The scanning function appears to be activated and the scanner light comes on, but no image is produced onscreen.

In these cases, a diagnostic package such as MSD can be quite helpful in spotting and correcting the conflicts.

The scanning function appears to be activated and the scanner light comes on symptom can also apply to the DMA channel of the interface card conflicting with the DMA setting of another card. Select another DMA channel for the card and software.

If the scanner light does not come on when trying to scan, there are two possible causes of the problem. First, the system may not meet minimum system requirements for the particular scanner being used. If this is the case, the minimum system requirements of the scanner should be researched to make sure that it can be used with the system. If the system requirements are not met, either return the scanner to the supplier or upgrade the system to meet the minimum specifications for the scanner. The second possible cause of the problem is that the I/O address setting conflicts with another card in the computer. Try other address settings on the adapter card and in the software.

If the problem seems to be a hardware problem, check the quick things as described in Chapter 5. Make sure the power to the scanner is plugged in and turned on. Exchange the signal cable for a new one, if available. Refer to the scanner's user's guide for troubleshooting hints.

Summary

The first half of this chapter examined standard I/O port assignments in a PC-compatible system. At this point, you should be able to identify the type of port being used from a description of its structure. In addition, you should be able to use the troubleshooting information provided for each port type to isolate and correct typical port problems in a PC-compatible system.

The second half of the chapter dealt with common input devices used with PCs. You should be able to describe the operation of each type of input device. You should also be able to list the types of devices normally associated with each I/O port type. As with the I/O ports in the first half of the chapter, you should be able to use the troubleshooting information provided for each input device to isolate and correct problems typically associated with that device.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each point and can perform each task listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercise

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 20 - Keyboard Problem Isolation and 21 - Mouse Operations.

Review Questions

1. What type of integrated circuits is normally found in a PC-compatible keyboard?
2. How are scan codes transmitted from the keyboard to the system?
3. Is a keyboard an input device, an output device, or an I/O device?
4. What type of connector is normally found on PC-compatible keyboards?
5. If a compatible Windows mouse driver is not present, what action should be taken?
6. If swapping the keyboard for a known good one does not clear the problem, what is the second most likely cause of the problem?
7. If the keyboard works properly in DOS, but will not work in Windows, where should you check for problems? What type of problem would you be looking for?
8. Why would the troubleshooting routine for a suspected keyboard problem begin with the "exchange the keyboard for a known good one" instruction?
9. What portion of the system board is dedicated to the operation of the keyboard?
10. Describe two types of touch screens.

11. How does the color of a grayscale scanner's light source affect its operation?
12. Does the color of the scanner's light source affect line art and text scans?
13. What is the most common problem encountered when adding a scanner to an 8-bit machine?
14. What type of input device detects the electron beam passing by on the CRT screen?
15. Describe the mechanical operation of an opto-mechanical mouse.
16. If a mouse works under DOS, but not in Windows, what area should be checked?
17. Describe the operation of a common light pen.
18. What type of device is normally associated with a 15-pin, male D-shell connector?
19. What advantage does a trackball have over a mouse? In what type of application is this useful?
20. Describe how the wiring of a mouse to a COM port is different than that of other common serial devices.
21. What type of device is a mouse?
22. List four major types of key switches.
23. Describe how the term `debouncing` applies to keyboards.
24. Name two types of joysticks.
25. The input device most widely used with microcomputers is __________.

Review Answers

1. A microprocessor-like intelligent keyboard encoder. Advancements in LSI chip technology have produced single-device keyboard encoders that contain all the circuitry necessary to interface the computer to an electronic keyboard. For more information, see the section titled "Keyboard Encoding."

2. Each time the keyboard encoder receives a valid key closure from the matrix, it generates two serially coded characters. The first is a distinct 8-bit code (referred to as a scan code) that corresponds to the key closure. For more information, see the section titled "PC-Compatible Keyboards."

3. An input device. Data only moves out to the keyboard during initialization, to program it for operation. For more information, see the section titled "The Input Buffer."

4. The most common connector is a 5-pin DIN connector. However, 6-pin mini-DINs and SDL connectors are also used. For more information, see the section titled "Interfacing and Connection."

5. Try a generic driver and then contact the mouse maker for an updated Windows driver. For more information, see the section titled "Mouse Software Checks."

6. The keyboard receiver circuitry on the system board. For more information, see the section titled "Keyboard Hardware Checks."

7. A problem with the Keyboard Type entry in the Window's Setup window. For more information, see the section titled "Software Checks."

8. The keyboard is external, easily detached for exchange, and relatively inexpensive. For more information, see the section titled "Keyboard Hardware Checks."

9. The intelligent keyboard controller IC. For more information, see the section titled "Intelligent Keyboard Controllers."

10. Conductive membrane panels and infrared matrix panels. For more information, see the section titled "Touch-Sensitive Screens."

11. The intensity of the reflected light is influenced by the color of the scanning light. This intensity difference will affect how the human eye perceives the image. For more information, see the section titled "Handheld Scanners."

12. No. These types of objects are just black and white. Therefore, only two intensities are returned to the detector. For more information, see the section titled "Handheld Scanners."

13. Finding an available expansion slot for the adapter card. For more information, see the section titled "Flat-Bed Scanners."

14. A light pen. For more information, see the section titled "Light Pens."

15. The mouse's trackball turns the x and y displacement capstans. The capstans each drive perforated wheels. As the wheels turn, light shining through the perforations is chopped into light pulses that are detected and applied to the mouse encoder chip. The encoder converts the pulse information into a serial data stream that can be interpreted by the mouse driver software. For more information, see the section titled "Mouse/Trackballs."

16. Check the mouse driver entry in the Window's Setup window. For more information, see the section titled "Mouse Software Checks."

17. As the electron beam from the CRT passes by the light pen's lens, the beam creates a pulse that is detected and fed to the light pen register of the CRTC on the video adapter card. This value is compared to the values in the CRTC's cursor position registers to determine the x and y position onscreen where the beam was detected. For more information, see the section titled "Light Pen Operation."

18. A joystick or game paddle. For more information, see the section titled “Game Port.”

19. It does not require a surface to move across. For more information, see the section titled “Mouse/Trackballs.”

20. The mouse uses only four of the RS-232 pins for operation. For more information, see the section titled “Mouse Communications.”

21. A pointing device. For more information, see the section titled “Mouse/Trackballs.”

22. Hard contact switches, dome switches, capacitive switches, and magnetic core switches. For more information, see the section titled “Switching Techniques.”

23. Debouncing is a time delay incorporated between the first electrical contact of the switch and the taking of a sample. This allows intermittent spikes generated as the key contacts come near each other to be ignored. For more information, see the section titled “Keyboard Encoding.”

24. Analog and digital joysticks. For more information, see the section titled “Joysticks and Game Paddles.”

25. The alphanumeric keyboard. For more information, see the section titled “Keyboards and Keypads.”

Chapter 8
Magnetic Storage

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Differentiate between different types of hard- and floppy-disk drive types.
- State reasons for the popularity of magnetic disks as computer data storage systems.
- Describe the format or organization of a typical soft-sectored hard or flexible disk.
- Describe the components of a typical sector.
- Describe how 1's and 0's are actually written on a magnetic surface.
- Identify the major physical blocks of the disk drive unit.
- Explain why the DOS software is so important to the operation of a disk drive.
- List the events that occur during a track-seek operation.
- List the events that occur during a typical disk drive write operation.
- List the events that occur during a typical disk drive read operation.
- Describe how data is read back from the disk.

continues

- Differentiate between common connecting cables (for example, SCSI, IDE, and FDD).
- Describe steps to troubleshoot FDD problems.
- Describe steps to troubleshoot HDD problems.
- Discuss the different RAID advisory levels and apply them to given applications.
- Install IDE and EIDE devices, including setting master/slave/single designations.
- Install and configure single and complex SCSI device chains.
- Establish proper addressing and termination for SCSI devices to avoid conflicts and problems.

Introduction

Nearly every microcomputer includes some type of mass information storage system that allows it to store data or programs for an extended period of time. Unlike primary memory devices, which are fast and have a relatively low storage capacity, mass storage systems are usually slower and possess much larger storage potential. These systems represent an acceptable alternative to the IC RAM/ROM devices used on system and video cards. Like the ROM devices, these systems must be able to hold information even when the computer is turned off. On the other hand, they are like RAM devices in that their information can be updated and changed often.

The most widely used mass storage systems have typically involved covering some medium with a magnetic coating.

These include flexible Mylar disks (floppy disks), rigid aluminum hard disks, and various widths of flexible Mylar tape as illustrated in Figure 8.1. The information to be stored on the medium is converted into electromagnetic pulses, that in turn are used to create tiny, positive and negative magnetized spots on the magnetic surface. To retrieve or read the information back from the surface, the storage system needs only to detect the spots and decode them. The stored information can be changed at any time by re-magnetizing the surface with the new information.

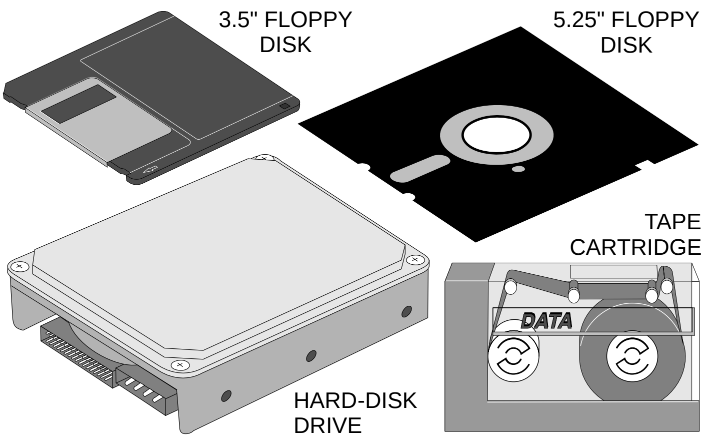


Figure 8.1

Typical magnetic storage systems.

Magnetic Disks

Objective

Magnetic disks resemble groove-less phonograph records, and fall into two general categories: high-speed hard disks and slower flexible disks. Data bits are recorded serially in concentric circles, called tracks, around the disk. Because the tracks toward the outer edge of the disk are longer than the inner tracks, all tracks are divided into an equal number of equal-size data blocks called sectors. Therefore each block of data has an address, which is the combination of its track number and its sector number. Because each sector can be accessed for a read or write operation as fast as any other sector on the disk, disk memories are classified as direct access memory.

The tracks of the disk are numbered, beginning with 00, from the outer edge of the disk, inward. Each side of the disk may hold 80 or more tracks, depending on the type of disk and the drive being used. When disks are stacked, such as in a hard-disk drive, all the tracks having the same number are referred to collectively as a cylinder. The number of sectors on each track runs between 8 and 50, also depending on the disk and drive type.

In some floppy-disk systems, the sectors are identified by holes along the outer or inner periphery of the disk. This is referred to as hard sectoring because hardware is used to identify the sectors, by physically counting the holes as the disk rotates. In other flexible- and hard-disk systems, track and sector address information is contained in a track/sector identification code recorded on the disk. This method of address specification is known as soft sectoring, because the sector information is written in software. Figure 8.2 depicts a typical soft-sectored track and sector arrangement. PC-compatible systems use soft-sectored disks.

Each sector is separated from the previous and following sectors by a small gap of unrecorded space. A typical sector contains 256 (2^8) or 512 (2^9) bytes, but on some systems this value may range as low as 128 (2^7) or as high as 1024 (2^{10}) bytes per sector. The most common sector size in the IBM-compatible/DOS world is 512 bytes. Within these confines, the sector is segmented, beginning with an ID field header, which tells the controlling circuitry that an

ID area containing the physical address information is approaching. A small `data field header` precedes the actual data field. The data field is followed by a `postamble,` containing error-checking and correcting codes for the data recorded in the sector.

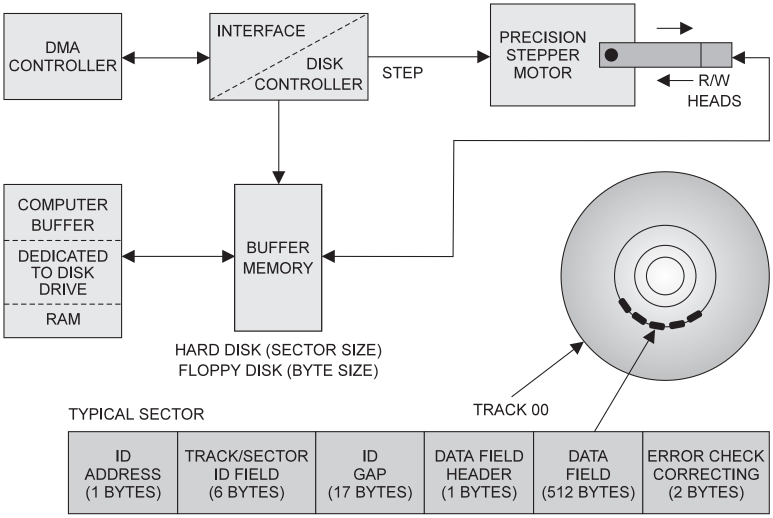


Figure 8.2

Disk tracks and sectors.

In their original conditions, hard disks and soft-sectored disks, as well as magnetic tapes, are blank. The system must prepare them to hold data. The system's disk drive control circuitry accomplishes this by writing track/sector identification and gap locations on the disk, in a process known as `formatting`. This leads to some confusion when disk storage capacity is specified. Capacity may be stated for either formatted or unformatted conditions. Obviously, the capacity of an unformatted disk is greater because no gap or ID information has been added to the disk.

Reading and Writing on Magnetic Surfaces

Objective

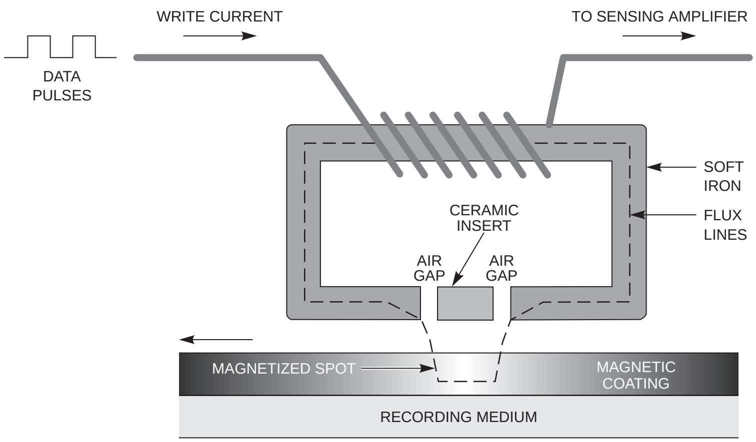
Data is read from or written to the disk one sector at a time. To perform a read or write operation, the address of the particular track and sector to be accessed is applied to a stepper motor, which moves a read/write (R/W) head over the desired track. As the desired sector passes beneath the R/W head, the data transfer occurs. Information read from, or to be written to, the disk is usually held in a dedicated part of the computer's RAM memory. The system then accesses the data from this memory location at microprocessor-compatible speeds.

The R/W head consists of a coil of wire wrapped around a soft iron core, as depicted in Figure 8.3. A small air gap in the core rides above the magnetic coating on the disk's surface. Data is written on the disk by pulsing the coil with a surge of current, which produces magnetic lines of flux in the soft iron core. At the air gap, the lines of flux dip down into the disk's magnetic coating because of its low reluctance (compared to air). This, in turn, causes the magnetic domains in the recording surface to align themselves in a direction dictated by the direction of current flow in the coil. The magnetic domains of the surface can assume one of three possible states, depending on the direction of current flow through the R/W head:

- Unmagnetized (randomly arranged domains)
- Magnetized in a positive direction
- Magnetized in a negative direction.

Figure 8.3

A typical R/W head.



Data is read from the disk in a reversal of this process. As the magnetized spots on the surface pass by the head, changes in magnetic polarity induce lines of flux into the iron core. This, in turn, induces a small voltage in the coil. This voltage is sensed and amplified to the proper digital logic levels by the drive's read circuitry. There is one significant difference between reading and writing on magnetic surfaces. Writing on the surface records successive zones of positive or negative flux, but Reading can only detect flux changes (that is, the boundaries between zones of differing polarity).

Data Encoding

Objective

Although it may seem logical to just represent 1s and 0s with positive and negative magnetic spots, the reality is that this brute force recording method is very ineffective and error prone. To compensate for these drawbacks, several encoding methods have been devised to represent 1's and 0's.

An early example of encoding was the frequency modulation (FM) method that used two recording frequencies to modulate the data. An advanced FM version called modified FM (or MFM) cuts out half the pulses and doubles the density of the data on the disk. This method is used to record double-density floppy disks in PC-compatible systems. Additional advancements in data storage are gained by using group coded recording (GCR) methods. This method uses selective bit patterns to encode blocks of data on the disk. The representative bit patterns are established so that they maximize the magnetic relationship between 0's and 1's to gain the best response from the R/W heads.

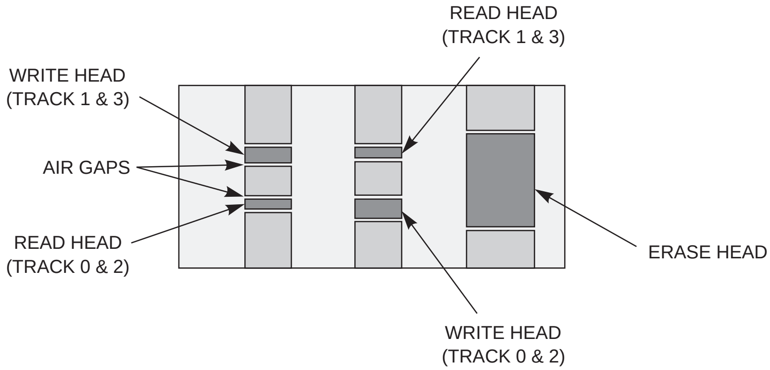
Head Construction

R/W heads are constructed in many different ways, depending on their intended application. R/W heads intended for use with disks are usually small, and electrically simple, compared to R/W heads used with tape drives. They normally use ferrite cores and a common winding for both reading and writing. Conversely, tape heads may have independent read and write cores, and multiple record/playback channels.

Quite often, the R/W head for data recording contains two independent cores, one designed for writing and the other designed for reading. The read gap is placed downstream from the write gap so that data being written to the surface can be read immediately and checked for errors. Some types of R/W heads contain as many as five air gaps, as shown in Figure 8.4. A gap may be only 30–100 micro-inches, but this is where all the work of recording and playback occurs.

Figure 8.4

A multiple-gap R/W head.



Multiple-gap R/W heads are generally used with magnetic tape operations; disks use single-gap heads. An exception to this trend is the R/W head associated with floppy disks. Floppy-disk heads contain a R/W gap followed by a `trim-erase gap`. The erase gap is used to trim the fringes of the recorded data to improve tracking. This type of head is also referred to as a "tunnel," or "straddle-erase" head.

The minimum distance between two successive flux reversals, called the "flux density," is measured in `flux-changes-per-inch` (fci) or `flux-reversals-per-inch` (frpi). These terms are a measurement of the number of bits of data that can be stored in a given area of media. Newer R/W head technologies have been developed to provide very high-density recording capabilities (15,000 fci).

Contact versus Noncontact Recording

Depending on the nature of the magnetic medium being read from or written to, the R/W head may ride directly on the medium's surface (`contact recording`), or it may "fly" slightly above it on a thin cushion of air created by the moving surface (`noncontact recording`). Hard disks, whether fixed or removable, must use flying heads; flexible media (tapes and disks) generally use contact recording.

Hard disks must unavoidably use noncontact heads that fly over the medium. The extremely high speed of the medium and the thin and fragile nature of its magnetic oxide coating make almost any contact between the R/W head and the disk surface a cause of considerable damage to both the head and the disk. Such contact is known as `head-to-disk interference` (HDI), or just a `crash`. Recent advancements, such as smaller and lighter R/W heads and ever harder damage-resistant disk surfaces, have lowered the possibilities of damaging crashes somewhat. Because the medium is dimensionally stable and spins at a high rate of speed, the data density associated with hard disks is relatively high. Complex head architectures are not used with this medium.

Flexible media, such as floppy disks and tape, expand and shrink with temperature and humidity variations. This causes the data tracks on the media to migrate in terms of track-location accuracy of the R/W head. To compensate for such shifting, the R/W heads ride directly on the media's surface, the track density is kept low, and the heads are made more complex to create special zones in the track construction, which compensate for some of the misalignment due to shifting.

Disk Drive Operations

Objective

The basic organization of both hard- and flexible-disk drives is similar in many respects. Both have drive spindles, which are actuated by precision synchronous motors, and one or more movable R/W heads positioned by a digital stepper motor or voice coil. Both systems have intelligent control circuitry to position the R/W head, and facilitate the transfer of information between the disk and the computer's memory. Figure 8.5 depicts the major components of the disk drive's electronics systems.

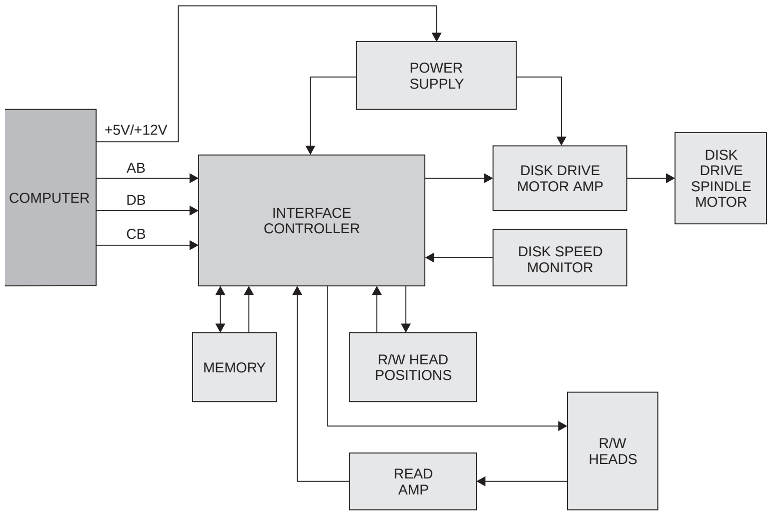


Figure 8.5

Disk drive components.

The heart of the disk drive's circuitry is the `disk drive controller`. The controller is responsible for providing the necessary interfacing between the disk drive and the computer's I/O channel. This consists of decoding the computer's instructions to the disk drive and generating the control signals that the disk drive must have to carry out the instruction. The controller must also convert back and forth between the parallel data format of the computer's bus and the serial format of the disk drive. In addition, the controller must also accurately position the R/W heads, direct the read and write operations, check and correct data received from the processor, generate error-correction codes for outbound data, and mark the location of defective sectors on the disk, closing them to use. After all these responsibilities, the disk controller must also provide addressing for the tracks and sectors on the disk, and control the transfer of information to and from the internal memory.

Because of the high data-transfer rates associated with disk systems, a DMA controller is used to permit the disk drive to access the computer's primary memory without the intervention of the CPU. In this manner, the disk drive can exchange data with the computer's main memory, while the microprocessor is busy at some chore that doesn't require the use of the address and data bus lines.

The extreme complexity of these responsibilities and the speed at which they must be performed usually dictate the presence of a specialized, onboard microprocessor, referred to as a `microcontroller`, to control the operation of the drive. Microcontrollers are actually `dedicated microprocessors` that have been designed expressly to control the operation of the peripheral. As a matter of fact, the first microprocessors were never actually marketed as such because they were developed as `peripheral control units` (PCUs) for discrete computers. Only in their second generation were their processing capabilities realized and brought to full use as microprocessors.

In floppy-disk drives, and hard drives designed around older interface standards, this controller is located on the drive's options adapter card. Drives that use newer interfacing methods have their controllers `embedded` directly on the disk drive.

With hard-disk drives, the presence of the onboard microcontroller is usually accompanied by onboard RAM and ROM, as well as any number of other support chips. The drive's RAM capabilities may be as simple as a single, 8-bit buffer register, or as complex as an array of RAM memory chips that can hold an entire track of information. This memory may be an integral part of the drive's intelligent controller, or it might consist of discrete memory devices.

Hard drives normally have read-only memories that contain an addition to the system BIOS program. This extension historically comes into play after the BIOS startup routine has checked for bootable files in floppy-disk drive A:, and not found them. Other support devices in the drive might include clocks (timers), decoders, drive-motor controllers, read and write amplifiers, shift registers, counters, write-protect circuits (on floppy drives), and power-supply circuits.

One piece of software that is extremely important to the operation of both hard- and floppy-disk drives is the DOS `File Management System` (FMS). Essentially, DOS keeps track of information stored on the disk by creating the `file allocation table` (FAT) on the disk. This table specifies where information is stored and what disk space remains available for new data. DOS matches the name assigned to a file with the blocks in the FAT, and then refers to a map called a `directory`, which tells the operating system how to get to a specific piece of data. In addition, the operating system provides disk-utility programs to perform functions such as copying and editing files, keeping track of unusable (bad) sectors, and aiding in the formatting of new disks.

Groups of logically related sectors are typically linked together into clusters (referred to as allocation units by DOS) on the disk. If a file on a disk requires several sectors of storage space, for example, DOS will look at the disk's FAT and try to store the parts of the cluster in logically sequential sectors. This strategy takes into consideration the read and repositioning times involved in moving between sectors. This storage method improves the speed at which the total cluster of information can be accessed.

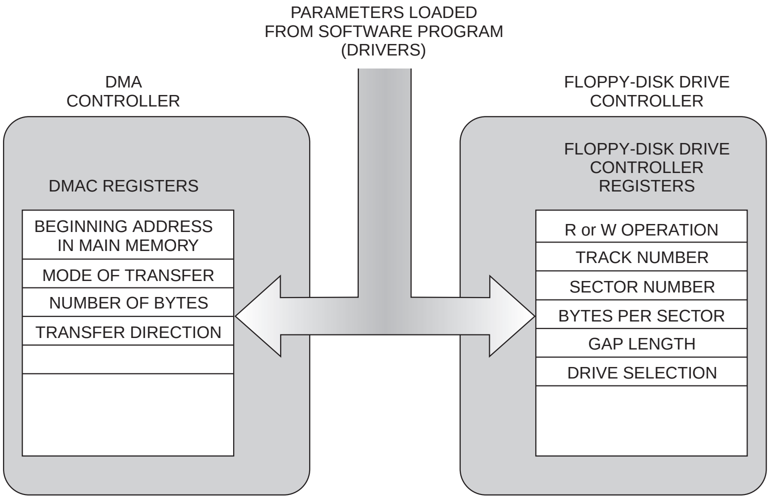
Initialization

A copy of the disk's directory and file allocation table are written into a dedicated portion of primary memory when the disk is booted up to the system. To transfer a file from main memory to the disk (a write operation), the operating system sends the disk drive controller a write command and initializes its control registers with parameters such as sector length and gap lengths. It also specifies the track and sector number where writing will begin. It obtains this information by referring to the FAT and finding the address of the next available sector, as shown in Figure 8.6.

The DOS software driver program plugs these parameters into the proper registers within the controller chip. It also sets up the DMA controller with information about where the starting address of the data to be written from main memory is, how long the block of data is (the number of bytes to be transferred), and in what mode the data transfer is to be performed (block or byte-by-byte). After this has been accomplished, the drivers set up the DMA and disk drive controllers for interaction, and arms them to begin transferring the data.

Figure 8.6

Initialization.



Track Seek Operations

Objective

The controller enables the drive and produces a burst of step pulses to position the R/W head over the proper track. The controller accomplishes this by keeping a record of the current track location of the drive's R/W heads in one of its internal registers. The controller compares the contents of the current location register to the track number specified by the write command and decides which direction the head must be moved. It then issues a direction signal to the drive on its `Direction` line, and begins producing step pulses on its `Step` line. Each pulse on this line causes the drive unit to move the R/W heads one track over, in the direction specified by the direction signal. The value in the current track location register is also increased or decreased (depending on the direction of movement) by a factor of one for each step pulse. When the value of the present location register matches the track number specified for the write operation, the step pulses cease and the R/W heads settle over the desired track. The positioning of the R/W heads is illustrated in Figure 8.7.

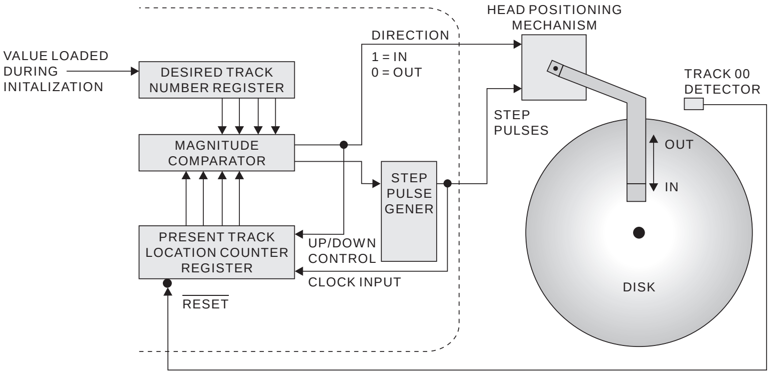


Figure 8.7

A track-seek operation.

Write Operations

Objective

When the head is over the track, the controller begins looking for the proper sector by reading the sector headers. The controller is actually waiting for a unique set of flux transitions to occur, which match the starting sector number stored in its registers. After the match is found, `write-splice` occurs (the controller changes from read to write operation) during the header gap, and data transfer and serialization begins.

The drive controller requests a byte of data (`DREQ`) from the DMA controller, which places the byte on the data bus and also sends a `DACK` signal to the disk controller. The disk controller obtains the byte (assuming byte-by-byte transfer) from the data bus, encodes it into the proper form, serializes it, and applies it to the write channel of the R/W head, as illustrated in Figure 8.8.

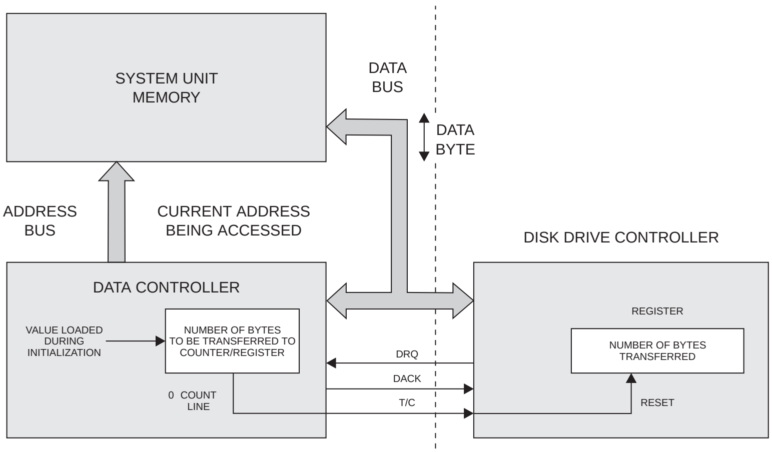


Figure 8.8

Data transfers.

The transfer continues byte-by-byte until the controller requests the last byte from the DMA controller. Counters in both the DMA controller and the disk-drive controller count the number of bytes transferred. They do this by decreasing the initialized value of the number of bytes to be transferred by a factor of one, each time a byte is transferred. When the counter in the DMA controller reaches its terminal count (0), it sends a `terminal count` (T/C) signal to the disk drive controller. If the counter in the disk drive controller does not equal zero at this time, an error flag is created in the disk-drive controller, and the system must attempt to write the data again.

Data transfers in hard-disk drives must occur at a much quicker pace than those in floppy-disk drives. For this reason, hard-disk drives use onboard RAM buffers to allow transfers to occur in block form rather than byte-by-byte. The buffer is small, usually only large enough to hold one sector of data, although some manufacturers use larger memories. When the controller begins serialization, the data is obtained from the buffer rather than through the DMA channel.

In addition to writing the data, the controller must also generate the sector postamble. At the end of the data field, the controller generates the postamble containing the error-detection and correction codes. The sector's `preambles` are written on the disk when it is formatted.

If the data from the computer requires more than one sector to be written, as it usually does, these logically related sectors may not be located sequentially on the disk. When data is transferred a block at a time, some time will be required to process each sector of data. To give the drive time to process the information, logically sequential sectors are `interleaved` (separated) by a fixed number of other sectors. This way, the motion of the disk is moving the second sector into position to be written (or read), while the drive is processing the preceding sector of information. A common `interleaving factor` is eight sectors between logically related sectors on a floppy drive; a factor of three on older, hard-disk systems; and a factor of one on newer systems.

Read Operations

Objective

When a read operation is performed, the operating system goes to the directory, either in main memory or the directory track on the disk, to get the starting track/sector address of the data to be read. This address is loaded into the disk controller and a track-seek operation is performed. The R/W head is stepped to the desired track, and the proper head is selected by the disk-drive controller. After a few milliseconds delay, to allow the R/W head to settle over the track, the operating system gives the disk controller the command to read the desired sector. The controller begins reading the sector ID headers, looking for the assigned sector.

When the sector is identified, the preamble is read and `bit-sync` is established. As the preamble is read, the controller synchronizes the data separator with the incoming bit stream from the disk drive. At the beginning of the data field, a `data start marker` initiates `byte-sync`, which coordinates the first bit of the first data byte with the controller's internal circuitry. At this point, the controller begins dividing the incoming bit stream into 8-bit words for transmission to the system unit.

After byte-sync has been established, the drive begins reading the data through the controller. The controller decodes the bit stream from its coded form (FM, MFM, GCR, and so on) and shifts it into an onboard shift register in 8-bit chunks. Hard-disk drives deliver the bytes to the onboard buffer memory; most floppy disks set up a DMA request and transfer the data bytes unbuffered into the computer's main memory. The transfer may continue over multiple sectors or tracks until an `end-of-file marker` is encountered, indicating that the entire file has been transferred.

Floppy-Disk Drives

The discussion of general disk-drive operations applies to both hard and floppy-disk drives alike. However, the physical construction and operation of the drives are quite different. The FDD is an exposed unit with an opening in the front to allow the floppy disk to be inserted and removed. In addition, the R/W heads are open to the atmosphere and ride directly on the surface of the disk. Older 5 1/4-inch units used a locking handle to secure the disk in the drive. A spring-loaded assembly ejected the disk from the drive when the handle was rotated. Newer 3 1/2-inch units have ejection buttons. Table 8.1 provides a comparison of the two different floppy-disk drive types using the disk formats possible for each.

Table 8.1

Comparisons of floppy-disk drive standards.

Diameter	Density	Capacity	Tracks	Sectors
3 1/2”	DD	720 KB	80	9
	HD	1.44 MB	80	18
5 1/2”	DD	360 KB	40	9
	HD	1.2 MB	80	15

Data moves back and forth between the system’s RAM memory and the floppy-disk surface. Along the way, it passes from the system RAM, to the `floppy-disk controller` (FDC), through the floppy-disk drive signal cable and into the floppy-disk drive’s `analog control board`. The analog control board converts the data into signals that can be applied to the drive’s read/write heads which, in turn, produce the magnetic spots on the disk surface.

In the original PCs and XTs, the FDC circuitry was located on the FDD controller card. In AT-compatible systems, it migrated onto a Multi I/O card along with parallel, serial, game, and HDD control ports. With Pentium-based systems, all this circuitry has been integrated into the system board.

The circuitry on the floppy-disk drive unit is usually distributed between two printed circuit boards: the analog control board and the drive’s `spindle motor control board`.

FDD Controller

Objective

In older PCs, the FDD control function was performed by a 765 FDC controller chip and a discrete digital control port register. In newer units, the FDC control function is provided by the floppy-disk controller (FDC) portion of the integrated I/O ASIC. This chip can be located on an MI/O card, or on the system board. To remain PC-compatible, the FDC registers and signal definitions used must remain identical to those of the 765 FDC and control port register. In any case, the FDD controller provides a programmable (intelligent) interface between the system unit and the floppy-disk drive unit. Figure 8.9 depicts a block diagram of the floppy-disk drive control circuitry.

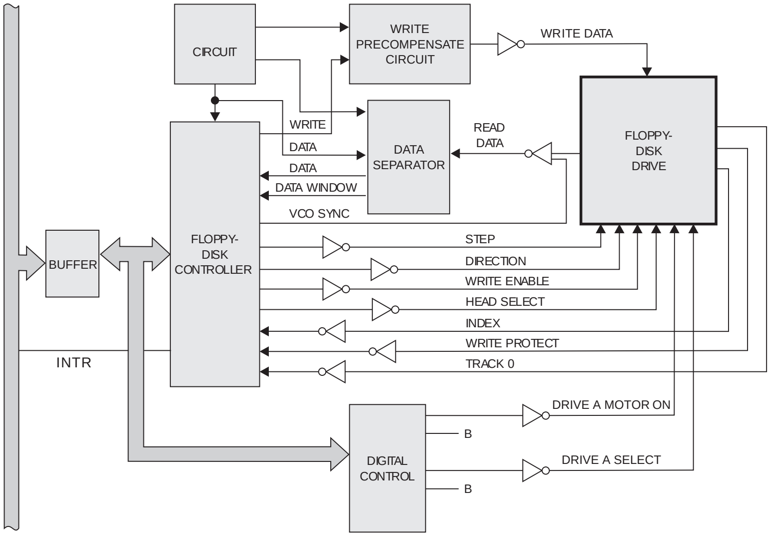


Figure 8.9

The floppy-disk drive adapter.

Under direction of the DOS system, the FDC divides the 3 1/2-inch floppy disk into 80 tracks per side, with nine or eighteen 512-byte sectors per side. This provides the system with 737,280 (720 KB) or 1,474,560 (1.44 MB) total bytes of storage on each disk. Table 8.2 lists the operating specifications for a typical 3 1/2-inch floppy-disk drive unit.

Table 8.2

FDD drive specifications.

Drive Unit Part	DSSD	DSHD
Tracks	80	80
Heads	2	2

Drive Unit Part	DSSD	DSHD
Sectors per track	9	18
Bytes per sector	512	512
Formatted capacity	720 KB	1.44 MB
Unformatted capacity	1 MB	2 MB
Rotational speed (RPM)	300	300
Recording density (bits/inch)	8717	17,432
Tracks per inch	135	135
Transfer rate unformatted (Kbps)	250	500

The 765-compatible FDC can carry out at least 15 different commands, such as `read sector`, `write sector`, `read track`, `seek`, and `format track`, under the direction of the disk operating system software. It supports both double-density and high-density (MFM) recording formats, and performs all the data decoding functions for the drive. It has the capability to control two drives (drives A and B) simultaneously, with any mixture of sizes and densities. In addition, the FDC performs all data synchronization and error-checking functions to ensure reliable storage and recall of data.

The ASIC provides the additional circuitry required to complete the entire FDD controller/interface section. The additional circuitry consists of the address-recognition circuitry, a clock signal generator to produce master and write clock signals, and a read data separator circuit to remove the data bits from the other information written in a sector. The FDC governs the operation of the data separator.

The ASIC also supplies interface signals that allow it to be connected to microprocessor systems with or without DMA capabilities. In most systems, however, the FDC operates in conjunction with the system's DMA controller and is assigned to the `DRQ2` and `DACK2` lines. In operation, the FDC presents a `DRQ2` request for every byte of data to be transferred. In addition, the disk drive controller is assigned to the `IRQ6` line. The FDC generates an interrupt signal each time it receives a read, write, or format command from the system unit. An interrupt will also be generated when the controller receives a `Ready` signal from one of the disk drive units. Figure 8.10 depicts a block function diagram of the FDC.

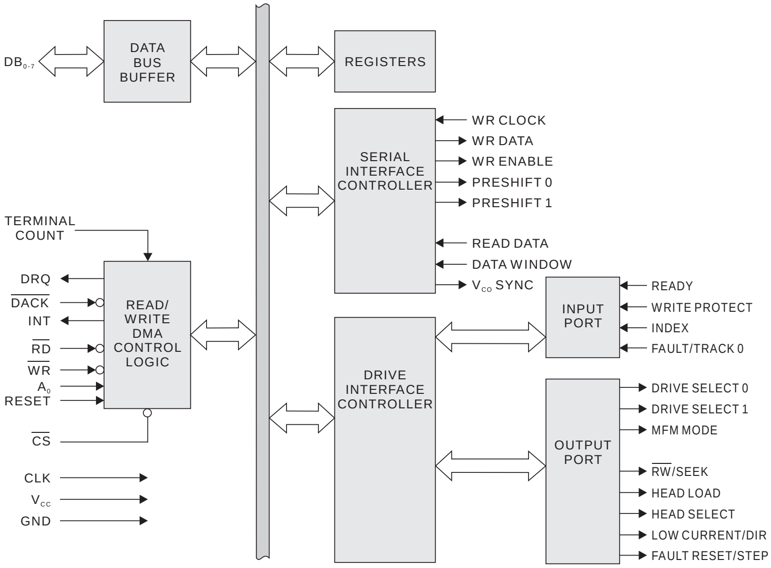


Figure 8.10

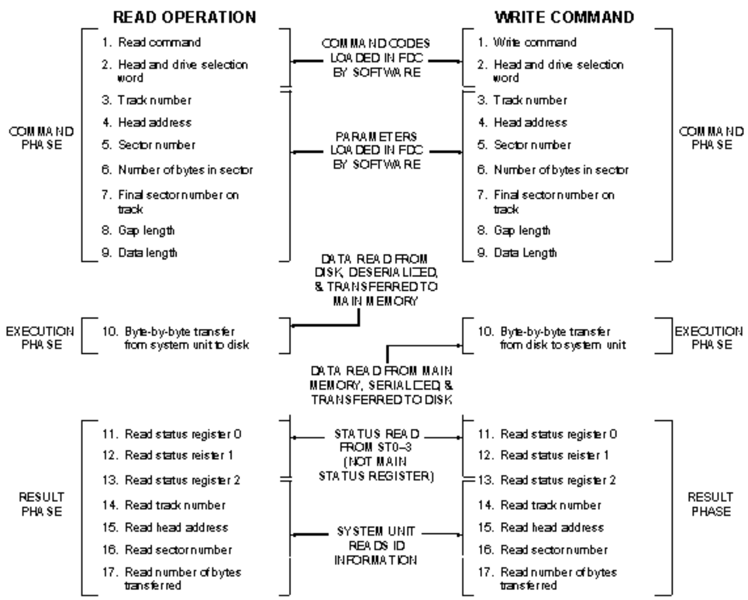
The FDC block diagram.

Of particular interest is the FDC's internal register set. It contains two registers that can be addressed by the system unit's microprocessor—an 8-bit `main status register`, and a stack of 8-bit registers called the `data register`. Only one of these data registers can be accessed at a time. Data bytes are written into, or read out of, the data register to program the controller, or to obtain results after the completion of an operation. The main status register contains status information about the drive and can be accessed by the system at any time.

The operation of the FDC occurs in a three-phase cycle: a `command phase`, an `execution phase`, and a `result phase`. During the command phase, the system unit initiates an action by writing a multibyte instruction and all related data into the FDC's data register. At the completion of the command phase, the FDC enters the execution phase, where the FDC generates the timing and control signals necessary for the drive to carry out the specified command. After the instruction has been carried out, the FDC enters the result phase, where the disk drive's status information is placed in the main status register. Figure 8.11 shows the sequence in which information is moved into the FDC's data register during read sector and write sector commands. Although the sequence is identical for these two commands, other instructions will occur in different sequences and with other parameters.

Figure 8.11

FDC R/W operations.



Digital Control Port Register

Operating in parallel with the FDC is the `digital control port register`. This register is used to control the selection of drive units A and B, the drive motors, and the adapter's FDC Reset, DMA Request (`DRQ2`), and Interrupt Request (`IRQ6`) functions. This is a write-only register, selected when the system unit applies an address of 3F2 to the adapter, with the `IOW` line asserted. The various bits of the control port register are used for drive/motor selection, `DRQ` and `IRQ` enabling circuitry, along with the FDC Reset bit.

Read Circuitry

Data sent to the FDC from the drive is actually a combination of two signals: the data, and the clocking information used to record it. This concept is easier to envision for FM-coded data, where each bit time is marked by the presence of a clock pulse. But

MFM-encoded data possesses clocking information embedded in the serial bit stream. To properly read the data, it must be broken into two parts: the clocking information and the data bits. This is the function of the `data separator` circuitry.

The data separator must synchronize the rate of the serial bit stream coming from the drive unit and the clocking of the FDC's internal circuitry (bit-sync). After synchronization is achieved, the separator creates a `data window` signal, used by the FDC to differentiate between the clocking information and the data bits in the bit stream. The FDC uses this window to reconstruct the data bits from the bit stream.

In addition to recovering the data bits from the bit stream, the FDC must break the data bits into 8-bit words for transfer to the system unit (byte-sync). This requires that the system unit's DMA controller service the FDC approximately every 15 microseconds for MFM-recorded data. If this servicing does not occur, the FDC will detect that a data register overrun error has occurred and set the `overrun error` flag in its main status register. This action terminates the data read command activities.

Figure 8.12 shows the disk drive adapter's read channel components. The MFM-encoded data stream from the disk drive passes through pin 30 of the disk drive connector. Next, the data stream is then applied to the data separator, where it is combined with the FDC's clock signal. From the two inputs, the data separator produces two outputs: the serial bit stream, and the `read data window` (`RDW`) signal. The `RDW` signal causes the FDC to sample the `read data` (`RDD`) line. During the window, the FDC loads the data bit on its read data input into the internal data register.

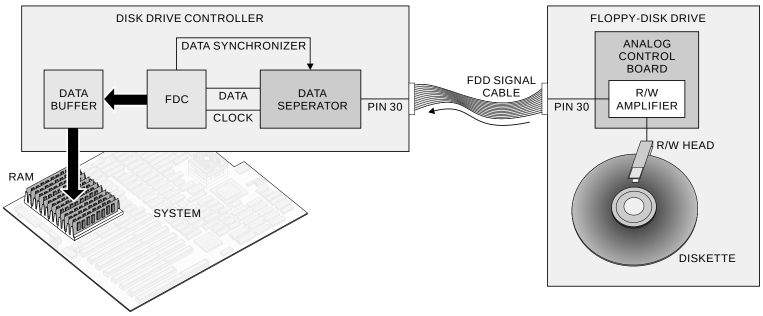


Figure 8.12

FDC read circuit.

Write Circuitry

Consider the disk drive adapter's write circuitry. When the system unit places data in the FDC to be written on the disk, the FDC serializes the data, codes the data in the designated form (FM or MFM), and adds error-detection and correction bits (CRCs) to the serial data stream. The data moves in encoded, serial format from the FDC's write data (WD) output, through a write precompensate circuit to the adapter's FDD connector at pin 22. The cable carries the WD and write enable (WE) signals to the disk drive unit.

The write precompensate circuit acts as a time delay to correctly position the data bits for proper read-back. This is necessary because a certain amount of data shift occurs during read-back. This shifting is predictable, so the FDC is programmed to precompensate the data stream before it is written.

The precompensation circuitry is only required for MFM-encoded data. MFM-encoded data does not contain a clocking pulse in its bit cell. Therefore it cannot tolerate large amounts of data-shifting when the data is read back from the disk.

Floppy-Drive Cables

Objective

A single ribbon cable is used to connect the system's floppy drive(s) to the disk-drive controller card. Generally, the cable has two edge connectors and two 34-pin, two-row BERG headers along its length. The edge connectors allow the cable to be connected directly to the printed-circuit board of a 5 1/4-inch FDD. The other end of the cable terminates in a 34-pin, two-row BERG header. A small, colored stripe normally runs along one edge of the cable, as illustrated in Figure 8.13. This is the pin #1 indicator stripe that marks the side of the cable that should be aligned with the #1 pin of the FDD adapter's connector and the disk drive's signal connector. The location of this pin is marked on the drive's printed-circuit board.

The floppy-disk drive connected to the 34-pin header (or edge connector) at the end of the cable will be assigned the drive A: designation by the system. A floppy drive connected to the edge connector in the middle of the cable will be designated for drive

B:. A small twist of wires between the A and B connectors reroutes key lines that determine which drive is which.

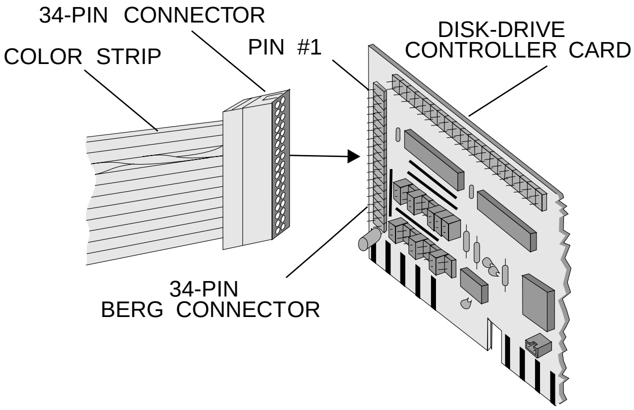


Figure 8.13

The FDD signal cable.

The 34-pin interface connection allows the FDC to control two, separate floppy-disk drive units. Figure 8.14 depicts the connections between the disk drive adapter, and the disk drive(s). The direction of signal flow between the drive(s) and adapter are indicated by the arrows.

Figure 8.14

FDD cable signal definitions.

DISK DRIVE

FDD ADAPTER

SIGNAL NAME	ADAPTER PIN NUMBER
GROUND (ODD NUMBERS)	1-33
DENSITY SELECT	2
UNUSED	4, 6
INDEX	8
MOTOR ENABLE A	10
DRIVE SELECT B	12
DRIVE SELECT A	14
MOTOR ENABLE B	16
DIRECTION	18
STEP	20
WRITE DATA	22
WRITE ENABLE	24
TRACK 0	26
WRITE PROTECT	28
READ DATA	30
SELECT HEAD 1	32
DISK CHANGE	34

All the adapter's signal lines are TTL-compatible. Furthermore, the functions of the lines are summarized as follows:

WRITE DATA (pin 22): For each high-to-low logic transition on this line, the disk drive stores a flux change on the disk. This action depends on the write enable line being activated.

READ DATA (pin 30): The selected disk drive places a pulse on this line for each flux change on the disk which passes under the selected R/W head.

WRITE ENABLE (pin 24): This line disables the drive's write circuitry unless it is active.

INDEX (pin 8): The selected drive applies a pulse to this line each time the index hole on the disk passes the index sensor. (1 pulse/revolution or approximately 300 pulses/min).

TRACK 0 (pin 26): When the R/W heads of the selected drive are positioned over track 0, the drive's track-00 sensor activates this line.

WRITE PROTECT (pin 28): The presence of a write-protected disk in the selected drive causes this line to be activated.

DRIVE A (pin 14): This line enables the drive unit attached as drive A: when active.

MOTOR A (pin 10): This line starts the A: drive spindle motor when activated, and stops it when the line returns to a high logic level.

DRIVE B and MOTOR B (pins 12 and 16): These lines are identical to the Drive A and Motor A lines, except they control the drive connected as drive B.

STEP (pin 20): During a seek operation, the FDC issues pulses on this line. The selected drive must move the R/W heads one track per pulse. The direction of movement is in accordance with the condition of the Direction line.

DIRECTION (pin 18): When this line is high, the selected disk drive moves the R/W heads one track away from the center of the disk for each pulse on the Step line. When low, the drive moves the heads one track toward the center for each pulse on the Step line.

SLCT HEAD 1 (pin 32): When this line is high, the upper R/W head (Head 0) of the selected drive is activated. When low, the lower head (Head 1) is activated.

DENSITY SELECT (pin 2): This line sets the write current level used for double-density or high-density disks. The controller outputs a low when double-density disks are detected, or a high when high-density disks are detected.

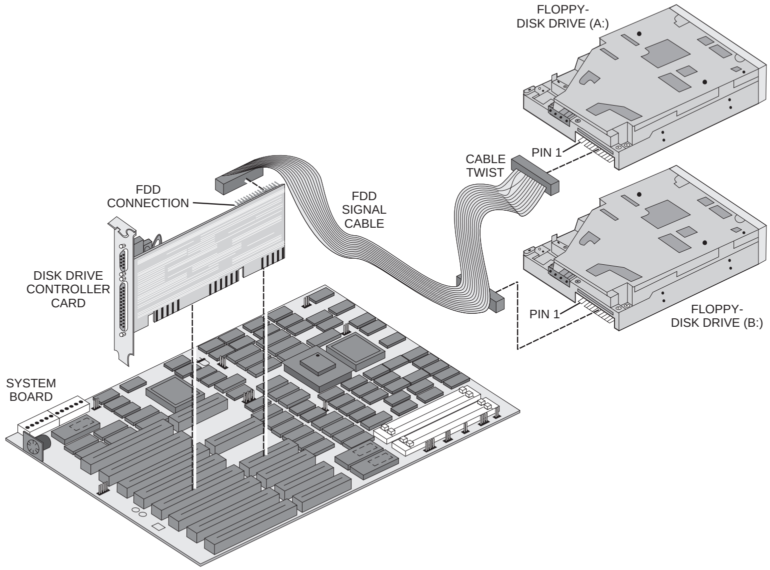
Installation

To install a floppy-disk drive, disconnect the system's power cord from the back of the unit. Slide the FDD into one of the system unit's open drive bays, and install two screws on each side to secure the floppy-disk drive to the system unit. If the unit is a 3 1/2-inch drive, and it is being installed into a 5 1/4-inch drive bay, you need to fit the drive with a `universal mounting kit`. These kits attach to the drive and extend its form factor so that it fits correctly in the 5 1/4-inch half-height space.

Connect the floppy drive signal cable to the 34-pin, FDD interface header (on an I/O card or system board). Then connect the signal cable to the floppy-disk drive, as illustrated in Figure 8.15. If the drive is the only floppy in the system, or intended to operate as the A: drive, connect the drive to the connector at the end of the cable. If it is being installed as a B: drive, attach it to the connector toward the center of the cable. On older floppy drives, the cable will connect to an edge connector on the drive's printed-circuit board. With newer units, the connection is made to a BERG connector. Note the orientation of the color stripe on both ends.

Figure 8.15

Connecting the floppy drive.



Connect one of the power supply's optional power connectors to the FDD unit. Check for the presence of a `drive select` (DS) jumper block on the drive's control board. This jumper is normally set to the DS0 position for drive-A operation. On Pentium and other types of all-in-one system boards, look for an `FDD enabling jumper`, and make certain that it is set correctly for the FDD installed.

Reinstall the system unit's power cord and boot up the computer. As the system boots, move into the CMOS Setup screens and configure the CMOS for the type of FDD just installed.

Troubleshooting FDD

Objective

Typical symptoms associated with floppy-disk drive failures during bootup include the following:

- FDD errors encountered during bootup.
- Front panel indicator lights visible, display present on the monitor screen, but there is no disk drive action and no bootup.
- An IBM-compatible 6xx error code is displayed.
- An FDD Controller Error message—indicating a failure to verify the FDD setup by the System Configuration file.

- The FDD activity light stays on constantly, indicating that the FDD signal cable is reversed.

Additional FDD error messages commonly encountered during normal system operation include the following:

- `Disk Drive Read Error` messages
- `Disk Drive Write Error` messages
- `Disk Drive Seek Error` messages
- `No Boot Record Found` message
- No boot record found—indicating that the system files in the floppy disk's boot sector are missing or have become corrupt.
- System stops working while reading a floppy disk—indicating that the contents of the disk have become contaminated.
- The drive displays the same directory listing for every disk inserted in the drive—indicating the FDD's Disk-Change detector, or signal line is not functional.A number of things can cause improper floppy-disk drive operation or disk drive failure. These items include the use of unformatted disks, incorrectly inserted disks, damaged disks, erased disks, loose cables, drive failure, adapter failure, system board failure, or a bad or loose power connector.

Information is written to and read from the floppy disks by the floppy-disk drive unit. This unit moves information and control signals back and forth between the disk drive controller and the surface of the disks. The information moves between the controller and the drive through a flat, ribbon cable. The small printed-circuit board, located on the drive unit, is called the `analog control board`. It is responsible for turning the digital information received from the adapter card into magnetic information that can be stored on the surface of the disk, and vice versa.

The floppy-disk drive controller is responsible for controlling the flow of information between the system board's circuitry and the disk drive. When the controller is mounted on an options adapter card, information and control signals pass between the card and the system board through the adapter card's edge connector and the system board's expansion-slot connector.

Basically three levels of troubleshooting apply to FDD problems: `configuration`, the `DOS level`, and the `hardware level`. No Windows-level troubleshooting applies to floppy-disk drives.

FDD Configuration Checks

Normally the only time a configuration problem occurs is when the system is being set up for the first time, or when a new option is installed. The other condition that will cause a configuration problem involves the system board's CMOS backup battery. If the battery fails, or has been changed, the contents of the CMOS set-up will be lost. After replacing the battery, it is always necessary to run the CMOS Setup utility to reconfigure the system.

While booting up the system to the DOS prompt, observe the BIOS FDD-type information displayed on the monitor. Note the type(s) of FDD(s) that the BIOS believes are installed in the system. With newer BIOS, it will be necessary to examine the advanced CMOS setup to check the bootup order. In these BIOS, the boot order can be set so that the FDD is never examined during startup.

The values stored in this CMOS memory must accurately reflect the type and number of FDDs installed in the system; otherwise an error will occur. These values can be accessed for change during the bootup procedure.

DOS Checks

If the FDD configuration information is correct, and a floppy-disk drive problem is suspected, the first task is to make certain that the system won't boot up from the floppy-disk drive if a disk with a known good boot file is in the drive. Try the boot disk in a different computer to see whether it works on that machine. If not, there is a problem with the files on the disk. If the disk boots up the other computer, it will be necessary to troubleshoot the floppy drive system.

If possible, run a diagnostic software program from the hard drive or a B floppy-disk drive. Try to use a diagnostic program that conducts a bank of tests on the FDD's components. Run the program's `FDD tests`, and perform the equivalent of the ALL tests function.

From the DOS level, it is also very easy to test the operation of the drive using a simple batch program. At the DOS prompt, type **Copy Con:FDDTEST.BAT**, and press the Enter key. On the first line, type the DOS **DIR** command. On the second line, type **FDDTEST**. Finally, press the F6 function key to exit, and save the program to the hard-disk drive.

This test program can be executed from the DOS prompt just by typing its name. When invoked, it will exercise the drive's R/W head-positioning motors and read channel signal processing circuitry. At the same time, the signal cable and the FDC circuitry will be tested.

FDD Hardware Checks

If a floppy-disk drive hardware problem is suspected, begin troubleshooting the hardware by removing all the externally connected devices from the system, except for the monitor and keyboard. Try to boot the system. If the system operates correctly with these options removed, it is safe to assume that one of them is the cause of the problem. To verify which external device is causing the problem, reconnect the devices, one at a time, until the problem reappears. The last device reinstalled before the problem reappeared is defective. Replace this item and continue reinstalling options, one at a time, until all the options have been reinstalled. If another failure occurs while reinstalling options, replace that option as well. Repair or replace the defective options device(s), and return the system to full service.

Note

> The process for isolating floppy-disk drive problems is reinforced and expanded in the accompanying Hands-On Lab Book in procedure 26.

If the system will not boot up with the external options removed, turn the system off, and remove the system unit's outer cover. Remove all the options adapter cards, except for the video and disk drive controller cards from the system board's expansion slots. Try to reboot the system.

If the system boots up with the internal options cards removed, it is reasonable to assume that one of them is the cause of the problem. Therefore you should reinstall the internal options, one at a time, until the problem reappears. As always, the last option reinstalled before the problem returned is defective. Repair or replace the defective options adapter as indicated, and return the system to full service.

If the system does not boot up with all the options removed, check the components associated with the floppy-disk drives. Begin by exchanging the floppy-disk drive with another one of the same type. If there is a second floppy-disk drive in the system, turn the computer off and exchange its connection to the floppy-disk drive's signal cable so that it becomes the A: drive. Try to reboot the system using this drive as A. Also check the floppy-disk drive's signal cable for proper connection at both ends. Check for a `drive select jumper` (DS) on the drive's printed-circuit board, and make certain that it is installed correctly.

Insert the bootable disk in the new A: drive, and turn the system on. If the system boots up, reinstall any options removed and replace the system unit's outer cover. Return the system to full service and repair the defective floppy drive accordingly.

If the system still refuses to boot up, turn it off and exchange the disk drive controller card (if present) with a known good one. Disconnect the disk drive's signal cable from the controller card, and swap the controller card with a known good one of the same type. Make certain to mark the cable and its connection point to ensure proper reconnection after the exchange. Reconnect the signal cable to the FDD controller.

Try to reboot the system with the new disk drive controller card installed. If the controller is built in to the system board, it may be easier to test the drive and signal cable in another machine than to remove the system board. If the system boots up, reinstall any options removed and replace the system unit's outer cover. Return the system to full service and return the defective controller card.

If the system still will not boot up or perform FDD operations correctly, check the disk drive cables for proper connection at both ends. If necessary, exchange the signal cable with a known good one. Finally, exchange the system board with a known good one.

Hard-Disk Drives

Although early PCs did not rely on hard drives, nearly every modern PC has at least one of them installed. After the original PC, the hard drive became standard equipment. The XT featured a 10 MB, MFM unit. Modern units feature drives that typically have storage capacities in the gigabyte range. Logically, the hard drive is organized as a stack of disks similar to a floppy disk. Each surface is divided into tracks, which are, in turn, divided into sectors. Each disk possesses a matching set of tracks on the top and bottom of the disk. The disks are stacked on top of each other and the R/W heads move in and out between them. Because there are corresponding tracks on the top and bottom of each disk in the stack, the HDD controller organizes them into `cylinders`. Cylinder 1 of a four-platter HDD, for example, would consist of track 1 of each disk surface. The cylinder concept is shown in Figure 8.16.

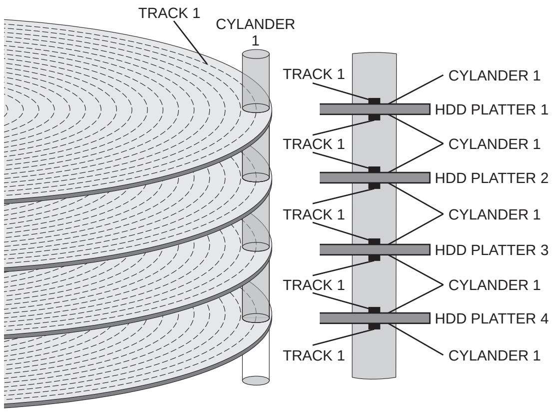


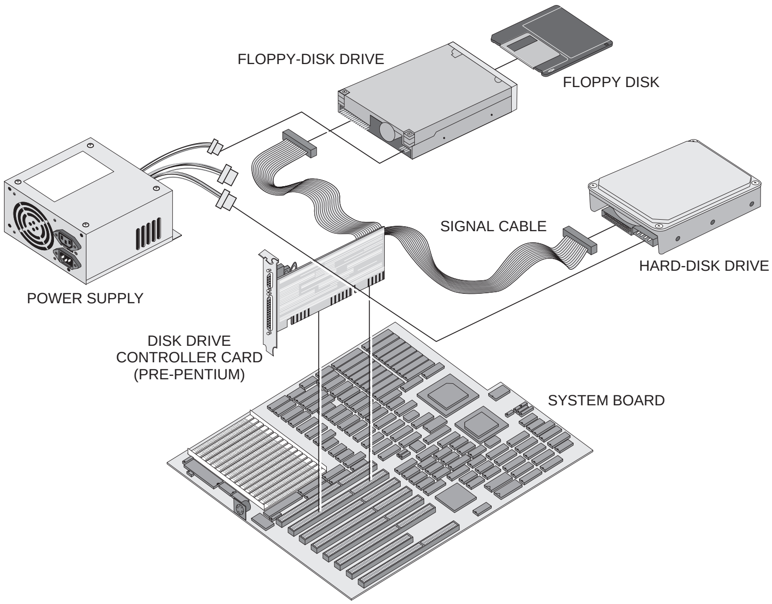
Figure 8.16

HDD cylinders.

The physical make-up of a hard-disk system is depicted in Figure 8.17. It involves a controller (either on an I/O card or built in to the system board), one or more signal cables, a power cable, and a disk drive unit. In some cases, floppy- and hard-disk drive signal cables may look similar. However, there are some slight differences in their construction that prevent them from being compatible. Therefore great caution must be used when installing these cables. Many skilled technicians have encountered problems by not paying attention to which type of cable they were installing with a particular type of drive.

Figure 8.17

Components of the HDD system.



The system's CMOS setup holds the HDD configuration settings. As with other configuration settings, these must be set correctly for the installed drive. Typical HDD information required for the CMOS setup includes the HDD's capacity, number of cylinders, number of R/W heads, number of sectors/track, amount of precompensation, and the track number to be used as the landing zone for the R/W heads when the drive is shut down. This information must normally be obtained from the drive manufacturer. Figure 8.18 gives typical HDD format information associated with a particular BIOS. Systems using other BIOS may have different values. Most BIOS tables also allow for a user-definable HDD entry, where the values are entered manually into the CMOS settings.

Figure 8.18

Typical HDD format values.

Type	Cylin	Heads	W-Pcomp	L-Zone	Capacity
1	306	4	128	305	10MB
2	615	4	300	615	21MB
3	615	6	300	615	31MB
4	940	8	512	940	63MB
5	940	6	512	940	47MB
6	615	4	FFFF	615	21MB
7	462	8	256	511	31MB
8	733	5	FFFF	733	31MB
9	900	15	FFFF	901	112MB
10	820	3	FFFF	820	21MB
11	855	5	FFFF	855	36MB
12	855	7	FFFF	855	50MB
13	306	8	128	319	21MB
14	733	7	FFFF	733	43MB
15	000	0	0000	000	00MB
16	612	4	0000	663	21MB
17	977	5	300	977	41MB
18	977	7	FFFF	977	57MB
19	1024	7	512	1023	60MB
20	733	5	300	732	31MB
21	733	7	300	732	43MB
22	733	5	300	733	31MB
23	306	4	0000	336	10MB
24	925	7	0000	925	54MB
25	925	9	FFFF	925	69MB
26	754	7	754	754	44MB
27	754	11	FFFF	754	69MB
28	699	7	256	699	41MB
29	823	10	FFFF	823	69MB
30	918	7	918	918	54MB
31	1024	11	FFFF	1024	94MB
32	1024	15	FFFF	1024	128MB
33	1024	5	1024	1024	43MB
34	612	2	128	612	10MB
35	1024	9	FFFF	1024	77MB
36	1024	8	512	1024	68MB
37	615	8	128	615	41MB
38	987	3	987	987	25MB
39	987	7	987	987	58MB
40	820	6	820	820	41MB
41	977	5	97	977	41MB
42	981	5	981	981	41MB
43	830	7	512	830	49MB
44	830	10	FFFF	830	69MB
45	917	15	FFFF	918	115MB
46	000	0	0000	000	00MB
47					

The other important disk drive specifications are access time, seek time, data-transfer rate, and storage capacity. These quantities designate how much data the drive can hold, how fast it can get to a specific part of the data, and how fast it can move it to the system.

Formatting

Unlike floppy drives, which basically come in four accepted formats, hard-disk drives are created in a wide variety of storage capacities. When the drive is created, its surface is electronically blank. To prepare the disk for use by the system, three levels of preparation must take place. These are, in order, the low-level format (below DOS), the partition, and the high-level format (or DOS).

A low-level format is very similar to a land developer sectioning off a field for a new housing development. The process begins with surveying the property and placing markers for key structures such as roads, water lines, and electrical service. The low-level format routine is similar in that it marks off the disk into cylinders and sectors, and defines their placement on the disk. In older device-level drive types (such as ST-506 and ESDI drives), the user is required to perform the low-level format. This procedure could be accomplished through the DOS Debug program by typing **G=C800:5** at the Debug prompt. In some units, the offset is 6 or 8 rather than 5. Many of the software diagnostic packages come with a low-level formatter program.

Most newer, system-level drive types (such as IDE and SCSI drives) come with the low-level format already performed. Therefore they do not require low-level formats, as do some of the older interface types. Attempts to perform low-level formats on IDE and SCSI drives may result in damage to the drive. This is not physical damage, but the loss of prerecorded bad track and sector information that would occur during a low-level format. The drive also contains alignment information used to control the R/W heads for proper alignment over the tracks. This alignment information would also be lost during a low-level format. If this occurs, it will normally be necessary that the drive be sent to the manufacturer to restore this information to the disk.

Logical and Physical Drives

Before a high-level format can be performed, the drive must be partitioned. The partition establishes the size and number of logical `drives` on the `physical drive`. By partitioning a drive, multiple operating systems can exist on the same drive. The oldest versions of MS-DOS (2.x, 3.x) imposed a limit on the size of a logical drive at 32 MB each—identified by different drive letters (C, D, E, for example).

As HDD technology steadily increased, the sizes of the physical drives eventually passed this limit. DOS version 4.0 raised the maximum size of a logical drive to 128 MB; version 5.0 raised it to 528 MB. Special disk management installation packages have extended the size limit of a logical drive up to 2 GB. The partitioning program for MS-DOS is named `FDISK`. Likewise, the FDISK utility in Windows 95 provides upgraded support for very large hard drives. The original version of Windows 95 set a size limit for logical drives at 2 GB. The FDISK version in the upgraded OSR2 version has extended the maximum disk partition size to 8 GB.

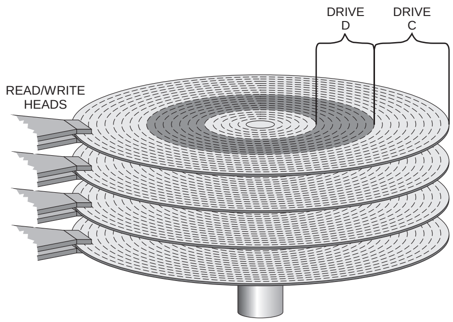
Even though newer versions of DOS allow partitions larger than 528 Mbytes, there is another factor that limits the size of disk partitions—the BIOS. The standard BIOS has a 504 Mbyte capacity limit. To overcome this, newer BIOS include an Enhanced mode that uses Logical Block Addressing (LBA) techniques to enable the larger partition sizes available through DOS and Windows to be used. This technique, known as Enhanced Cylinder, Heads, Sectors (ECHS), effectively increases the number of R/W heads the system can recognize from 16 to 256. The parameters of 1024 cylinders, 63 sectors/tracks and 512 bytes/sectors remains unchanged.

Figure 8.19 illustrates the concept of creating multiple logical drives on a single hard drive. This is normally done for purposes of organization and increased access speeds. Basically, DOS allows two partitions to be created on an HDD unit. The first, or `primary partition`, must exist as the C: drive. The system files must be located in this partition, and it must be set to "active" for the system to boot up from the drive. After the primary partition has been established and properly configured, an additional partition, referred to as an `extended partition`, is also allowed. However, the extended partition may be subdivided into 23 logical drives. The extended partition cannot be deleted if logical drives have been defined in it.

In local area and wide area networks (LANs and WANs), the concept of logical drives is carried a step further. A particular hard-disk drive may be a logical drive in a large system of drives along a peer-to-peer network. On the other hand, a very large, centralized drive may be used to create several logical drives for a server/ client type of network.

Figure 8.19

Partitions on an HDD.



The high-level format procedure is performed by the `FORMAT` command in the MS-DOS program. This format program creates the blank file allocation table and root directory on the disk. These elements tell the system which files are on the disk, and where they can be found. Never format a disk with an older version of DOS than is currently installed on the disk. The drive can actually be damaged from this action. Before reformatting a drive, use the DOS `VER` command to determine which version of DOS is currently in use.

HDD Interfaces

Objective

There have been four HDD interfaces commonly associated with microcomputers. These include two device-level interfaces and two system-level interfaces. The device-level interfaces are the ST-506/412 and Enhanced Small Device Interface (ESDI). These types of drives typically use a controller card that contains the system-level interface for the drive. These drive types require the user to perform all three levels of drive preparation.

The system-level interfaces—Integrated Drive Electronics (IDE) and Small Computer System Interface (SCSI)—place most of the controller circuitry on the drive itself. Therefore the system sees the entire HDD system as an attachment to its bus systems. As noted in the preceding section, these units also come with the low-level format already in place.

The following sections describe the major attributes of the four HDD interface types.

ST-506/412 MFM Interface

Some hard-disk drives, such as those used in 8088-based computers, use a two-cable arrangement to connect the drive to the options adapter card, as depicted in Figure 8.20. This type of drive is referred to as an MFM drive (because it uses MFM coding), and the physical connection is specified as an ST-506/412 interface. A similar type, called a `run-length limited` (RLL) drive, also uses this interface-connection scheme. These drives use the same physical equipment as the MFM drives, but employ `group-coded recording` (GCR) methods to extend the drive's storage capacity by 50%.

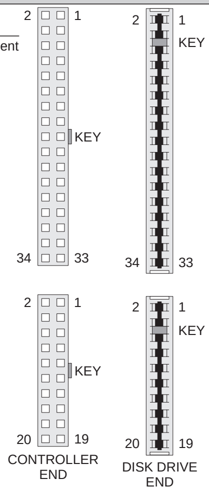
Figure 8.20

The ST-506-412 interface cable.

34-PIN SIGNAL CABLE

Pin	Description	Pin	Description
1	Ground	2	Reduced Write Current
3	Ground	4	Head Select 4
5	Ground	6	Write Gate
7	Ground	8	Seek Complete
9	Ground	10	Track 00
11	Ground	12	Write Fault
13	Ground	14	Head Select 1
15	Ground	16	Reserved
17	Ground	18	Head Select 2
19	Ground	20	Index
21	Ground	22	Ready
23	Ground	24	Step
25	Ground	26	Drive Select 1
27	Ground	28	Drive Select 2
29	Ground	30	Drive Select 3
31	Ground	32	Drive Select 4
33	Ground	34	Direction In

CONNECTOR ENDS



20-PIN DATA CABLE

Pin	Description	Pin	Description
1	DRV SLCTD	2	Ground
3	Reserved	4	Ground
5	Reserved	6	Ground
7	Reserved	8	Ground
9	Reserved	10	Reserved
11	Ground	12	Ground
13	Mfm WR Data	14	Mfm Write Data
15	Ground	16	Reserved
17	Mfm RD Data	18	Mfm Read Data
19	Ground	20	Ground

In the ST-506 interface, the smaller, 20-pin `data cable` is easily identified with the MFM or RLL hard drive. However, the larger 34-pin `signal cable` closely resembles the floppy-disk drive's signal cable, and can often be confused with it. The two cables are not interchangeable. The difference between them is found in the twisted wires located between the two edge connectors. On the floppy cable, the twist begins with the tenth wire in from the indicator stripe and involves seven wires (conductors 10–16). With the hard-disk drive's signal cable, the twist begins with the sixth wire in from the opposite side of the cable and involves five wires (conductors 25-29).

The drive unit connected to the edge connector at the end of the cable is normally designated by the system as the C: drive. The drive connected to the middle of the cable is usually configured as the D: drive. The addition of a second physical MFM drive to the system will require that a second 20-pin data cable be added the interface to accommodate it. There are also single-connector versions of the signal cable, called `straight-through cables`, available for use in systems that only require one HDD unit.

Enhanced Small Device Interface (ESDI)

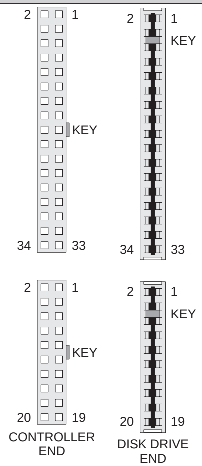
Objective

The `Enhanced Small Device Interface` (ESDI) is actually an improved ST-506 standard interface. This interface, shown in Figure 8.21, uses the same 20/34-pin, two-cable arrangement as the ST-506. The signal definitions and locations have been changed, however, so that the two interfaces are not compatible.

34-PIN SIGNAL CABLE

Pin	Description	Pin	Description
1	Ground	2	Head Select 3
3	Ground	4	Head Select 2
5	Ground	6	Write Gate
7	Ground	8	Config/ Status Data
9	Ground	10	Transfer ACK
11	Ground	12	Attention
13	Ground	14	Head Select 0
15	Ground	16	Sector Mark Found
17	Ground	18	Head Select 1
19	Ground	20	Index
21	Ground	22	Ready
23	Ground	24	Transfer Request
25	Ground	26	Drive Select 1
27	Ground	28	Drive Select 2
29	Ground	30	Drive Select 3
31	Ground	32	Read Gate
33	Ground	34	Direction In

CONNECTOR ENDS



20-PIN DATA CABLE

Pin	Description	Pin	Description
1	Drive Selected	2	Sector Mark Found
3	Seek Complete	4	Address Mark Enable
5	Reserved	6	Ground
7	Write Clock	8	Write Clock
9	Cartridge Chg	10	Read Ref Clock
11	Read Ref Clock	12	Ground
13	NRZ Write Data	14	NRZ Write Data
15	Ground	16	Ground
17	NRZ Read Data	18	NRZ Read Data
19	Ground	20	Index

Figure 8.21

The ESDI interface cable.

Unlike the ST-506 standard, where the controller has to be configured by the system for the type of drive it is being connected to, the ESDI interface calls for the drive to provide the controller with its configuration information. This information includes which type of drive it is and how many tracks and sectors it has.

The drive also provides bad track and sector information about itself. The information is installed on the disk at the factory by the manufacturer. These drives generally double the capacity and transfer rate of an equivalent MFM drive. Both of these improvements are accomplished primarily by increasing the number of sectors on each track to 34 (as opposed to 17 for most MFM drives).

Integrated Drive Electronics (IDE) Interface

Objective

The `Integrated Drive Electronics` (IDE) interface is a system-level interface. It is also referred to as an `AT Attachment` (ATA) interface. The IDE interface places most of the controller electronics on the drive unit. Therefore data travels in parallel between the computer and the drive unit. The controller circuitry on the drive handles all the parallel-to-serial and serial-to-parallel conversions. This allows the interface to be independent of the host-computer design.

An IDE drive stores formatting information on itself. This information is placed on the drive by its manufacturer and is used by the controller for alignment and sector sizing of the drive. The IDE controller uses a data separator, much like the FDD data separator, to intercept and isolate the raw data (format and actual information) coming from the R/W heads. This means that the information passed to the drive is composed of commands and data. The mixed data stream leads to much higher bandwidth requirements between the IDE drive and the host adapter. Although the data-transfer rate varies from model to model, IDE drives have rates several times greater than older ST-506 and ESDI designs.

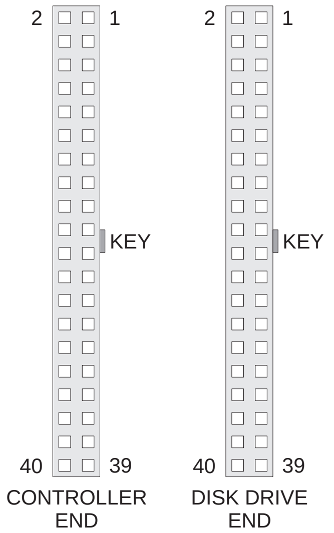
The IDE interface uses a single 40-pin cable to connect the drive(s) to the adapter card or system board. Its signal cable arrangement is depicted in Figure 8.22. In a system-level interface, the I/O card that plugs into the expansion slot is called a `host adapter` rather than a controller card. It should be apparent from the figure that the IDE host adapter is quite simple, because most of the interface signals originate directly from the system's extended bus lines.

Figure 8.22

The IDE signal cable.

PIN DESCRIPTION **CONNECTOR ENDS**

Pin	Description	Pin	Description
1	Reset	2	Ground
3	Data 7	4	Data 8
5	Data 6	6	Data 9
7	Data 5	8	Data 10
9	Data 4	10	Data 11
11	Data 3	12	Data 12
13	Data 2	14	Data 13
15	Data 1	16	Data 14
17	Data 0	18	Data 15
19	Ground	20	Unused
21	Unused	22	Ground
23	IOW	24	Ground
25	IOR	26	Ground
27	IOCHRDY	28	Bale
29	Unused	30	Ground
31	IRQ14	32	IOCS16
33	A1	34	PDAIG
35	A0	36	A2
37	HDCS0	38	HDCS1
39	SLV ACT	40	Ground



The host adapter basically serves three functions. These include providing the select signals to differentiate between a `single` drive system and the `master` and `slave` drives. It also provides the three least significant address bits (A0–A2) and the interface Reset signal. The HDCS0 signal is used to enable the master drive, and the HDCS1 signal is used to enable the slave drive.

Updated IDE specifications have been developed to allow for more than two drives to exist on the interface. This new specification is called `Enhanced IDE` (EIDE), or the `ATA-2` interface. Actually, the update covers more than just increasing the number of drives that can be accommodated. It also provides for improved IDE drivers, known as the `AT Attachment Packet Interface` (ATAPI), for use with CD-ROM drives, as well as new data-transfer methods.

The new standard adds single-word and double-word DMA transfer capability to the interface's standard Programmed I/O mode. The Single-word mode provides for one 16-bit word to be transferred during each DMA request cycle. In Multi-word mode, the data transfer is conducted in a DMA Burst mode until the `DREQ` line is deactivated, or until the DMA controller's terminal count is reached.

Most IDE drives come from the manufacturer configured for operation as a single drive, or as the master drive in a multidrive system. To install the drive as a second (or slave) drive, it is usually necessary to install, remove, or to move a jumper block. Some hosts disable the interface's `cable select` pin (pin 28) for slave drives. With these types of hosts, it is necessary to install a jumper for the `Cable Select` option on the drive. Consult the system's User's Manuals to see whether it supports this function.

In the MS-DOS system, the primary partitions of multiple IDE hard drives are assigned the first logical drive identifiers. If an IDE drive is partitioned into two logical drives, the system will identify them as drives C: and D:. If a second IDE drive is added as a slave drive with two additional logical drives, it will reassign the partitions on the first drive to be logical drives C: and E:; the partitions on the slave drive will be D: and F:.

Small Computer System Interface (SCSI)

Objective

The `Small Computer System Interface` (SCSI, often referred to as "scuzzy") standard, like the IDE concept, provides a true system-level interface for the drive. Nearly all the drive's controller electronics are located on the peripheral device. As with the IDE host adapter, the duties of the SCSI host adapter are reduced to mostly physical connection functions, along with some signal compatibility handling.

Using this arrangement, data arrives at the system interface in a form that is already usable by the host computer. This can be seen through the SCSI interface description in Figure 8.23. Note that the SCSI interface described in the figure only makes provisions for 8-bit parallel data transfers.

The SCSI interface can be used to connect up to seven diverse types of peripherals to the system. As an example, a SCSI chain could connect a controller to a hard drive, a CD-ROM drive, a high-speed tape drive, a scanner, and a printer. Additional SCSI devices are added to the system by daisy-chaining them together.

The input of the second device is attached to the SCSI output of the first device, and so forth.

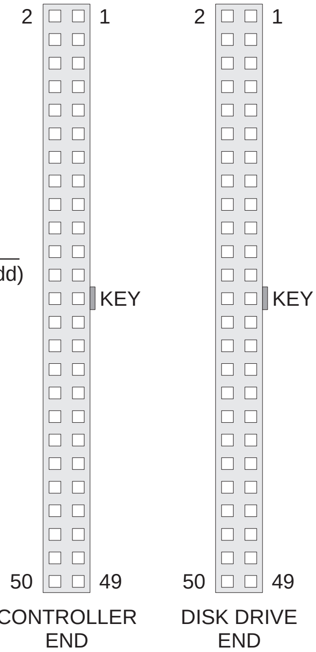
Figure 8.23

The SCSI interface cable.

PIN DESCRIPTION

Pin	Description	Pin	Description
1	Ground	2	Data 0
3	Ground	4	Data 1
5	Ground	6	Data 2
7	Ground	8	Data 3
9	Ground	10	Data 4
11	Ground	12	Data 5
13	Ground	14	Data 6
15	Ground	16	Data 7
17	Ground	18	Data Parity (Odd)
19	Ground	20	Ground
21	Ground	22	Ground
23	Ground	24	Ground
25	No Connection	26	No Connection
27	Ground	28	Ground
29	Ground	30	Ground
31	Ground	32	Attention
33	Ground	34	Ground
35	Ground	36	Busy
37	Ground	38	ACK
39	Ground	40	Reset
41	Ground	42	Message
43	Ground	44	Select
45	Ground	46	C/D
47	Ground	48	Request
49	Ground	50	I/O

CONNECTOR ENDS



In PC-compatible systems, the SCSI interface uses a 50-pin signal cable arrangement, which should not be confused with any other interface cabling arrangements. Although the SCSI interface defined here is the full standard, some manufactures may not utilize all the functions of the interface. The version of the SCSI interface used in the Apple Macintosh uses a variation of the standard that employs a proprietary, miniature 25-pin D-shell connector.

These types of variations create a hardware incompatibility between different SCSI devices. Likewise, some SCSI devices just will not work with each other because of software incompatibilities. In addition, SCSI devices may be classified as internal, or as external devices. An internal SCSI device has no power supply of its own and, therefore, must be connected to one of the system's options power connectors. On the other hand, external SCSI devices come with built-in or plug-in power supplies that need to be connected to a commercial ac outlet. Therefore when choosing a SCSI device, always inquire about compatibility between it and any other SCSI devices installed in the system.

When installing a SCSI device, addressing was originally set by jumpers on the host adapter card. In PnP systems, the BIOS configures the address using information obtained directly from the card during bootup. Unlike other HDD types, SCSI hard drives are not configured as part of the system's CMOS setup function. This is because DOS and Windows 3.x never included support for SCSI devices. Therefore SCSI drivers must be loaded during bootup before the system communicates with the drive. Windows 95 does, however, offer SCSI support. SCSI drives also require no low-level formatting. Therefore the second thing you do when installing a SCSI drive is to partition it.

The SCSI port can be daisy-chained to allow up to six external peripherals to be connected to the system. Although a total of eight SCSI device numbers is possible, only six are available for external devices. The SCSI specification refers to its SCSI controller as SCSI-7 (by default), and then classifies the first internal hard drive as SCSI-0.

Each SCSI device has either a SCSI number selection switch or configuration jumpers. The SCSI address setting must be unique for every device attached to the host adapter. If two devices are set to the same ID number, one or both of them will appear invisible to the system.

To connect multiple SCSI devices to a controller, all the devices, except the last one, must have two SCSI connectors. One for SCSI-In, and one for SCSI-Out. Which connector is which does not matter. If the device only has one SCSI connector, however, it must be connected at the end of the chain.

Multiple SCSI adapters can be used in a single system. The first SCSI controller can handle up to seven devices. An additional SCSI controller can boost the system to support up to 14 SCSI devices.

The SCSI daisy chain must be terminated with a terminating-resistor network pack at both of its ends. Single-connector SCSI devices are normally terminated internally. If not, a SCSI terminator cable (containing a built-in resistor pack) must be installed at the end of the chain. SCSI termination is a major cause of SCSI-related problems. Poor terminations cause a variety of different system problems including failed system startups, hard drive crashes, and random system failures.

Maximum recommended length for a complete SCSI chain is 20 feet. Unless the cables are heavily shielded, however, they become susceptible to data corruption caused by induced noise. Therefore a maximum single segment of less than three feet is recommended. Don't forget the length of the internal cabling when dealing with SCSI cable distances. You can realistically count on about three feet of internal cable. Therefore reduce the maximum total length to about 15 feet. Inside the computer, the SCSI specification uses a 50-pin ribbon cable with BERG pin connectors.

An updated SCSI specification has been developed by the ANSI committee. This specification doubles the number of data lines in the interface and adds balanced, dual-line drivers that allow much faster data-transfer speeds to be used. This implementation is referred to as `Wide SCSI-2`. The specification expands the SCSI specification into a 16/32-bit bus standard and increased the cable and connector specification to 68 pins.

An additional improvement increased the synchronous data-transfer option for the interface from 5 MB to 10 MB. This implementation became known as `Fast SCSI-2`. Under this system, the system and the I/O device conduct non-data message, command, and status operations in 8-bit asynchronous mode. After agreeing on a larger or faster file-transfer format, they conduct transfers using an agreed-upon word size and transmission mode. The increased speed of the Fast SCSI specification reduced the maximum length of the SCSI chain to about 10 feet.

A third version, brought together both improvements and became known as `Wide Fast SCSI-2`. An expected update, referred to as `Ultra Wide SCSI`, or SCSI-3, makes provisions for a special high-speed serial transfer mode and special communications media, such as fiber-optic cabling.

The increased speed capabilities of the SCSI interfaces make them attractive for intensive applications such as large file servers for networks, and multimedia video stations. The EIDE interface is generally more widely used, however, because of its lower cost and nearly equal performance. The EIDE device also tends to be more compatible with DOS. Most SCSI adapters come with software drivers that must be installed to support them. Table 8.3 contrasts the specifications of the SCSI and IDE interfaces.

Table 8.3

SCSI/IDE specifications.

INTERFACE	BUS SIZE	#DEVICES	ASYNC. SPEED	SYNC. SPEED
IDE	16 bits	2	4 MB/s	- -
EIDE (ATA-2)	16 bits	4	4 MB/s	16 MB/s
SCSI (SCSI-1)	8 bits	7	2 MB/s	5 MB/s
WIDE SCSI (SCSI-2)	8/16/32 bits	7	2 MB/s	5 MB/s
FAST SCSI (SCSI-2)	8/16 bits	7	2 MB/s	10/20 MB/s
WIDE FAST SCSI (SCSI-2)	8/16/32 bits	7	2MB/s	10/20/ 40/ MB/s
ULTRA WIDE SCSI (SCSI-3) -	-	-	-	

HDD Upgrading

One of the key components in keeping the system up to date is the hard disk drive. Software manufacturers continue to produce larger and larger programs. In addition, the types of programs found on the typical PC are expanding. Many newer programs place high demands on the hard drive to feed information, such as large graphics files, or digitized voice and video, to the system for processing. Invariably, the system will begin to produce error messages that say that the hard drive is full.

The first line of action is to use software disk utilities to optimize the organization of the drive. These utilities, such as CHKDSK, SCANDISK, and DEFRAG are covered in detail in Chapter 13 - Preventive Maintenance. The second step, is to remove unnecessary programs and files from the hard drive. Programs and information that is rarely, or never, used should be moved to an archival media, such as removable disks or tape.

In any event, there may come a time when it is necessary to determine whether the hard drive needs to be replaced in order to optimize the performance of the system. One guideline suggests that the drive should be replaced if the percentage of unused disk space drops below 20%.

Another reason to consider upgrading the HDD involves its ability to deliver information to the system. If the system is constantly waiting for information from the hard drive, replacing it should be considered as an option. Not all system slow downs are connected to the HDD, but many are. Remember that the HDD is the mechanical part of the memory system while everything else is electronic.

As with the storage space issue, HDD speed can be optimized through software configurations such as a disk cache. However, once it has been optimized in this manner, any further speed increases must be accomplished by upgrading the hardware.

When considering an HDD upgrade, determine what the real system needs are for the hard drive. Also, determine how much performance increase can be gained through other upgrading efforts (check the System Board Upgrading section of Chapter 6), before changing out the hard drive. Finally, determine how much longer the unit in question is likely to be used before simply being replaced. If the decision to upgrade the HDD stands, ultimately, the best advice is to get the biggest, fastest hard drive possible. Don't forget to look at the fact that a different I/O bus architecture may add to the performance increase.

HDD Installation

The installation steps for HDD hardware parallel those of the FDD unit. However, the configuration and preparation of a typical hard-disk drive is much more involved than that of a floppy-disk drive. This procedure is divided into two subsections. The first section deals with installing the hardware of a typical hard-disk drive unit. The second portion moves through a typical formatting procedure that can be applied to most microcomputer HDD units. If a replacement HDD is being installed for repair or upgrading purposes, the data on the original drive should be backed up to some other media before replacing it, if possible.

Turn the computer off and remove the outer cover from the system unit. If you are installing a 3 1/2-inch drive in a 5 1/4-inch bay, attach mounting brackets to both sides of the drive unit.

Obtain the drive's "type" information and record it. In addition, obtain any bad track/sector information from the HDD unit and record it. Slide the HDD unit into an empty bay of the disk drive cage, and install the four screws (two on each side) that secure it in the disk drive cage, as illustrated in Figure 8.24.

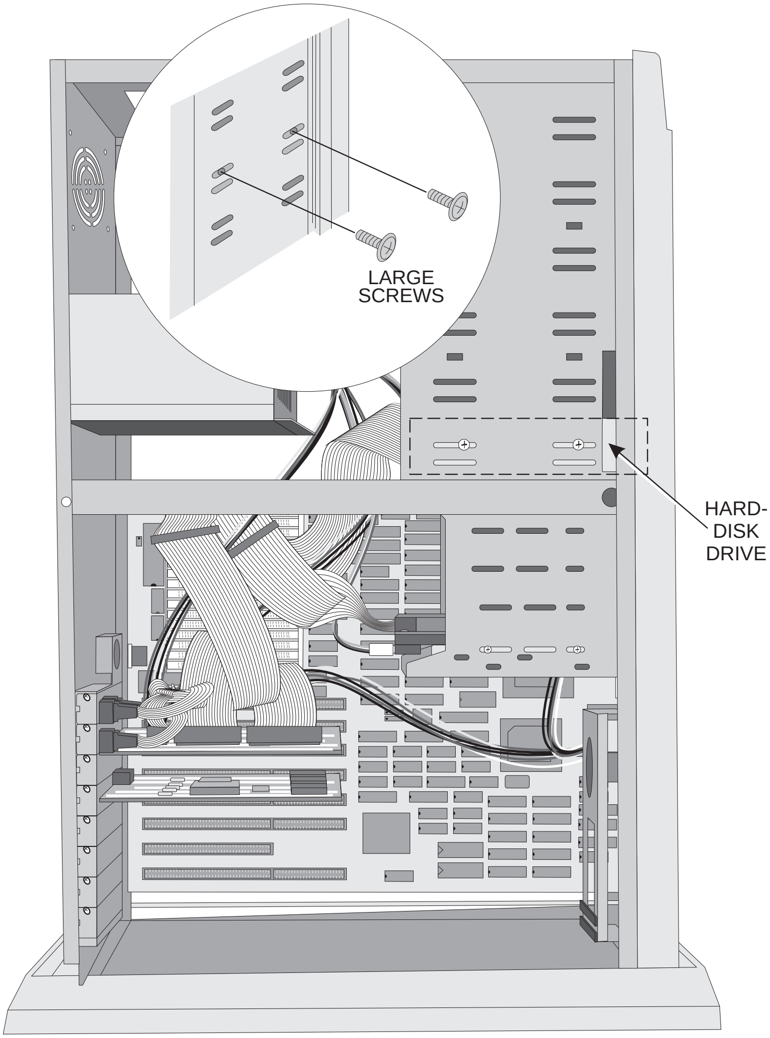


Figure 8.24

Securing the drive unit.

Connect the signal cable(s) to the HDD unit and to the controller. Make certain that the pin 1 indicator stripe on the cable aligns with the pin 1 position of the connectors on both the HDD unit and the controller. Proper connection of a single signal cable interface is depicted in Figure 8.25.

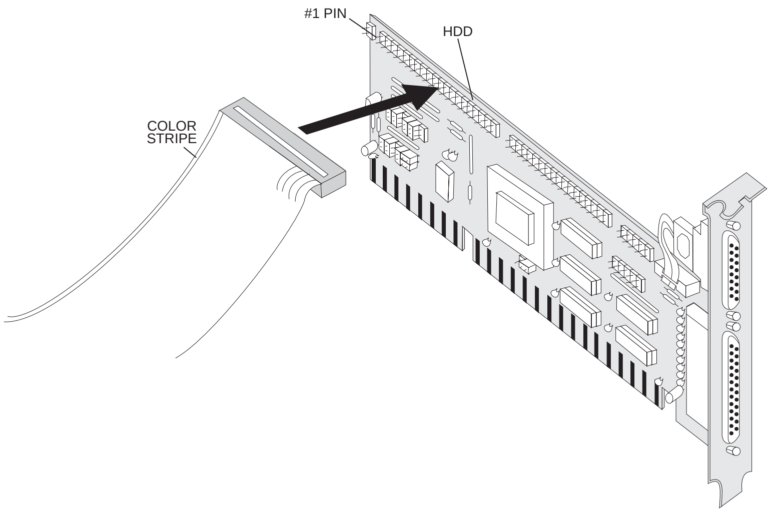


Figure 8.25

Connecting the HDD cable.

Turn the computer on and move into the BIOS's CMOS Configuration Setup screen. Move the cursor to the Hard Disk C type position, and scroll through the HDD selections until an entry matching the type information of the drive you are installing is found. Store this parameter in the CMOS configuration file by following the directions given in the screen menu.

The formatting procedure in older drive types is a function of the HDD's BIOS. System-level HDD interfaces, such as IDE and SCSI drives, have automatic low-level formatting routines already installed on themselves. Therefore these will not need a low-level format performed before they can be partitioned and high-level formatted.

It will be necessary to tell the formatting program what type of drive is being formatted, and what its parameters are. In newer drives, the information is retrieved from the system's CMOS setup information. In these cases, SCSI drives often require that the CMOS configuration be set to None Installed to operate correctly.

Other information you may be required to supply in older formatting programs is bad cylinder and sector information. This information can be found on the drive performance sheet that accompanies it from the factory. It can also typically be found on a sticker attached to the top of the drive. The technician must make certain to enter this information in the format specified by the program, and to answer any other questions asked by the program.

Turn the computer off and place a bootable disk in the A: drive. Turn the system on and install the operating system on the HDD. If a DOS or DOS/Windows 3.x system is being established, install DOS to the C:\DOS subdirectory. Use the `DIR` command to ensure that FDISK.COM is present in the DOS directory.

At the DOS prompt, type **FDISK** and press Enter to reach the DOS FDISK screen.

The following are the options that you can choose from:

- Create DOS Partition or Logical DOS Drive
- Set Active Partition
- Delete Partition or Logical DOS Drive
- Display Partition Information

Press 1 to choose Create a DOS Partition, and then press 1 to choose Primary DOS Partition. Finally, press Y to choose Maximum Size for this (Primary) Partition, and then press Enter to reboot the system and activate the partition.

If the system contains a master and a slave drive, the primary partition of the master drive will logically be the C: drive. The primary partition of the slave drive will be assigned drive D, with the extended partitions on the master and slave being assigned as drives E and F respectively.

Troubleshooting HDDs

Objective

Typical symptoms associated with hard-disk drive failures include the following:

1. The computer will not boot up when turned on.
2. The computer will boot up to a system disk in the A: drive, but not to the hard drive—indicating that the system files on the HDD are missing, or have become corrupt.
3. No motor sounds are produced by the HDD while the computer is running. (In desktop units the HDD should always run when power is applied to the system—this is not true of portables because of their advanced power-saving features).
4. An IBM-compatible 17xx error code is produced on the monitor screen.
5. An `HDD Controller Failure` message—Failure to verify hard-disk setup by System Configuration file error.
6. A `C: or D: Fixed Disk Drive Error` message—Indicates a hard-disk setup failure.
7. An `Invalid Media Type` message—Indicates the controller cannot find a recognizable track/sector pattern on the drive.
8. A No boot record found message (Non-system Disk or Disk Error, or Invalid System Disk)—Indicates that the system boot files are not located in the root directory of the drive.
9. The video display is active, but the HDD's Activity light remains on and no bootup occurs—indicating that the HDD's CMOS configuration information is incorrect.
10. An "Out of Disk Space" message—indicating that the amount of space on the disk is insufficient to carry out the desired operation.
11. A "Missing Operating System," "Hard Drive Boot Failure," or "Invalid Drive or Drive Specification," message—indicating that the disk's master boot record is missing, or has become corrupt.

12. A "No ROM BASIC Interpreter Not Found" message, followed by the system stopping—indicating that no Master Boot Record was found in the system. This message is produced only by PCs, XTs, and some clones.

13. A "Current Drive No Longer Valid" message—indicating that the HDD's CMOS configuration information is incorrect or has become corrupt.

Consider the relationship between the hard-disk drive and the rest of the system, and the control and signal paths through the system. Hard drive systems are very much like floppy-disk drive systems in structure: They have a `controller`, one or more `signal cables`, a `power cable`, and a `drive unit`. The troubleshooting procedure typically moves from setup and configuration to formatting, and finally into the hardware component isolation process. The system board is a logical extension of the components that make up the HDD system. Unless the HDD controller is integrated into it, however, the system board is typically the least likely cause of HDD problems.

Notice that unlike a floppy-disk drive, there is no Windows component to check with a hard-disk drive. Windows relies on the system's DOS/BIOS structure to handle HDD operations.

HDD Configuration Checks

While booting up the system to the DOS prompt, observe the BIOS's HDD-type information displayed on the monitor. Note the type of HDD(s) that the BIOS believes are installed in the system. The values stored in this CMOS memory must accurately reflect the actual format of HDD(s) installed in the system; otherwise an error will occur. Possible error messages associated with HDD configuration problems include `Drive Mismatch Error` and `Invalid Media Type`. These values can be accessed for change by pressing the Ctrl and Del keys (or some other key combination) simultaneously, during the bootup procedure.

If the HDD is used with a system board-mounted controller, check for the presence of an HDD enabling jumper on the system board. Make certain that it is set to enable the drive, if present. Check the drive to make sure that it is properly `terminated`. Every drive type requires a terminal block somewhere in the interface. On system-level drives, the master/slave jumper setting should be checked to make sure that it is set properly for the drive's logical position in the system.

Software Checks

If the HDD configuration information is correct and a hard-disk drive problem is suspected, the first task is to determine how extensive the HDD problem is. Place a clean boot disk in the A: drive and try to boot the system. Then, perform a DOS `DIR` command to access the C: drive. If the system can see the contents of the drive, the boot files have been lost or corrupted.

Look in the root directory for the system files (denoted by **..** and **...** entries) and the COMMAND.COM file. It is common to receive a `Disk Boot Failure` message on the monitor screen if this type of situation occurs. If the clean boot disk has a copy of the FDISK program on it, attempt to restore the drive's master boot record (including its partition information) by typing: `A>FDISK/MBR`. Providing that the hard disk can be accessed with the `DIR` command, type and enter the following command at the DOS prompt (with the clean boot disk still in the A: drive):

COPY A:COMMAND.COM C:

Type and enter the following line at the DOS prompt:

SYS C:

These two lines should copy the DOS system files from the DOS disk to the hard-disk drive. Turn the system off, remove the DOS disk from the A: drive, and try to reboot the system from the hard drive.

If the system boots up properly, check to see that the operating system commands are functioning properly. Also, check to see that all installed software programs function properly. Recheck the installation instructions of any program that does not function properly. Reinstall the software program if necessary.

Actually, five conditions will produce a `Bad or Missing COMMAND.COM` error message onscreen. The first condition occurs when the COMMAND.COM file cannot be found on the hard drive, and no bootable disk is present in the A: drive. The COMMAND.COM file is not located in the hard drive's root directory. This message is likely when installing a new hard drive or a new DOS version.

The message will also occur if the user inadvertently erases the COMMAND.COM file from the hard drive.

If the system cannot see the drive after booting to the floppy-disk drive, the complete HDD system will need to be examined. Attempt to run a diagnostic software program, if possible. Try to use a diagnostic program that conducts a bank of tests on the HDD's components. Run the program's HDD tests, and perform the equivalent of the ALL tests function.

HDD Hardware Checks

If the hard-disk drive cannot be accessed and the configuration settings are correct, it will be necessary to begin troubleshooting the hardware components associated with the hard-disk drive. As with other troubleshooting routines, begin by reducing the components of the system as much as possible. Remove all external options from the system, except the keyboard and video monitor.

Turn the system on. If it boots up properly, reinstall the options in the system one by one, until the problem reappears. The last options device reconnected to the system is defective. Repair or replace the defective options device and return the system to full service.

> **Note** The process for isolating hard-disk drive problems is reinforced and expanded in the accompanying Hands-On Lab Book in procedure 28.

If no external reason is found for the problems, it will be necessary to remove the outer cover from the computer and begin troubleshooting the internal components associated with the hard-disk drive. In a pre-Pentium system, the easiest component to check is the controller card that holds the HDD interface circuitry. Exchange the controller card with a known good one of the same type.

Make certain to mark both the floppy and hard drive control/signal cable(s) so as to identify their connection points and direction. This will help to ensure their proper reinstallation. Reconnect the disk drive signal cables to the new controller card.

Try to reboot the system from the hard drive. If the system boots up properly, check to see that all the DOS commands (`DIR`, `COPY`, and so on) are working properly. Also, check the operation of all the hard disk's software programs to make sure they are still functioning correctly. Reinstall any program that does not function properly.

If the system still won't boot up, recheck the System Configuration Setup to see that it matches the actual configuration of the HDD. Record the HDD values from the setup so that they will be available if a replacement drive needs to be installed.

The next logical step may seem to be to replace the hard drive unit. It is quite possible, however, that the hard drive may not have any real damage. It may just have lost track of where it was, and now it cannot find its starting point. In this case, the most attractive option is to reformat the hard disk. This action will give the hard drive a new starting point to work from. Unfortunately, it will also destroy anything that you had on the disk before. At the very least, attempting to reformat the drive before you replace it may save the expense of buying a new hard-disk drive that is not needed. Make certain to use the `/s` modifier, or repeat the `SYS C:` operation with the `Format` command, to restore the system files to the hard drive.

If the system boots up properly after reformatting the drive, reinstall any options removed from the system, replace the system unit's outer cover, and return the system to full service.

If not, check the HDD's signal cable for proper connection at both ends. Exchange the signal cable (or cables) for a known good one. Check the HDD's drive select jumper and master/slave/single jumper settings to make sure they set correctly. Exchange the HDD's power connector with another one from the power supply to make certain that it is not a source of problems.

If the reformatting procedure is not successful, or the system still won't boot from the hard drive, it will be necessary to replace the hard-disk drive unit with a working one. Disconnect the signal, control, and power cables from the HDD unit, and exchange it with a known good one of the same type. Reconnect the signal, control, and power cables to the replacement HDD unit.

If a similar computer is being used as a source of test parts, great care should be used in removing the HDD from its original computer and reinstalling it in the defective computer. With some interfaces, such as an MFM drive, it may be advisable to swap both the disk drive and the controller card together.

Try to reboot the system from the new hard drive. If no bootup occurs, reformat the new drive. Make sure that any information on the replacement drive has been backed up on floppy disks or tape before removing it from its original system.

If the system still won't boot up with a different HDD, swap the hard-disk drive's signal/control cables with known good ones. Make certain to mark the cables for identification purposes so that they will be reinstalled properly. Also, use a different power connector from the power-supply unit to make certain that the current connector is not a source of the problems.

Check the system's Configuration Setup to see that it matches the actual configuration of the new HDD. Check to see that all installed software programs function properly.

If the system reboots from the replacement drive without reformatting, replace the drive (either with the one you have just installed, or with a new one). Also, try reinstalling the original disk drive controller card to see whether it will work with the new drive.

If the system still boots up and operates properly, reinstall any options removed from the system. Replace the system unit's outer cover and return the system to full service. Reboot the system and reinstall all software programs to the new hard-disk drive. (See the Installation Guide from the software manufacturer.) Return the system to full service and return the defective controller card appropriately.

RAID Systems

Objective

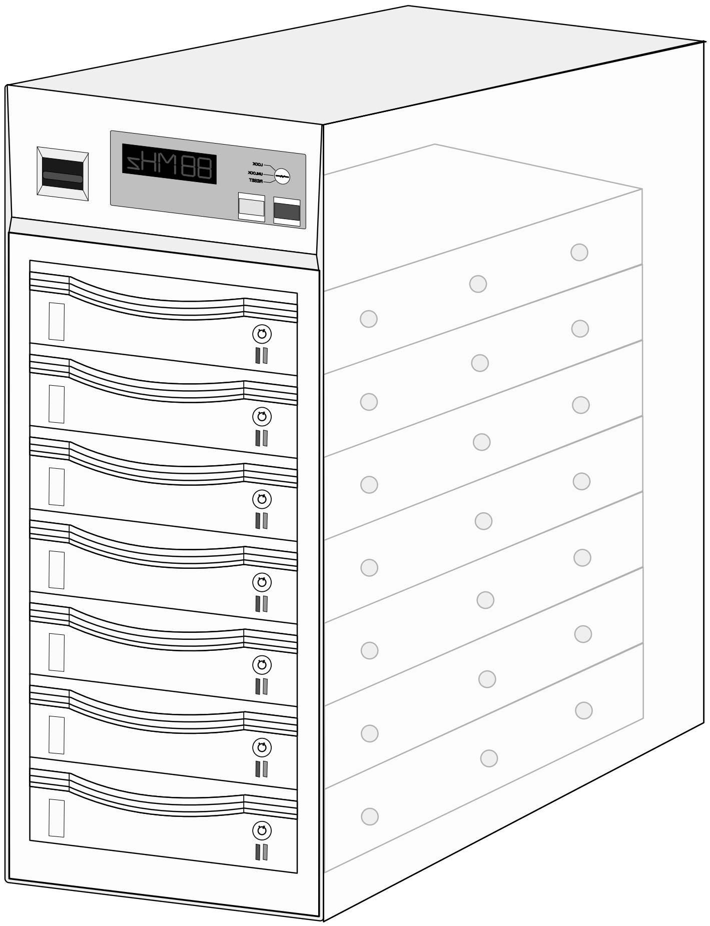
As applications push required storage capacities past available drive sizes, it becomes logical to combine several drives together to hold all the data produced. In a desktop unit, this can be as simple as adding an additional physical drive to the system. Wide and local area networks connect computers together so that their resources (including disk drives) can be shared. If you extend this idea of sharing disk drives to include several different drive units operating under a single controller, you have a `drive array`. Figure 8.26 shows a drive array.

Actually, drive arrays have evolved in response to storage requirements for local area networks. These are particularly useful in client/server networks, where the data for the network tends to be centrally located and shared by all the users around the network.

In the cases of multiple drives within a unit, and drives scattered around a network, all the drives assume a different letter designation. In a drive array, the stack of drives can be made to appear as a single, large hard drive. The drives are operated in parallel so that they can deliver data to the controller in a parallel format. If the controller is simultaneously handling eight bits of data from eight drives, the system will see the speed of the transfer as being eight times faster. This technique of using the drives in a parallel array is referred to as a *striped drive array*.

Figure 8.26

A drive array stack.



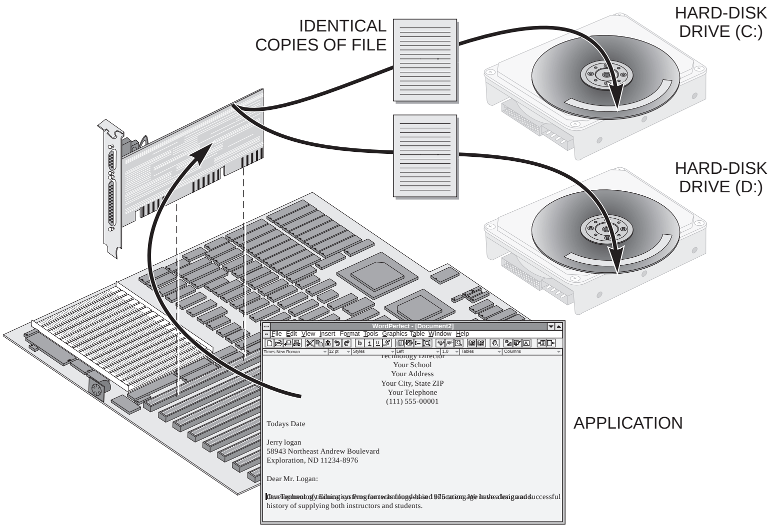
It is also possible to just use a small drive array as a data backup system. In this case, referred to as a `mirrored drive array`, the drives are each supplied with the same data. In the event that the data from one drive is corrupted, or one of the drives fails, the data is still safe. Both types of arrays are created through a blend of connection hardware and control software.

The most common drive arrays are `Redundant Arrays of Inexpensive Disks` (RAID) systems. There are five levels of RAID technology specifications given by the `RAID Advisory Board`.

`RAID 1` is a redundancy scheme that uses two, equal-sized drives, where both drives hold the same information. Each drive serves as a backup for the other. Figure 8.27 illustrates the operation of a mirrored array used in a RAID 1 application.

Figure 8.27

Operation of a mirrored array.

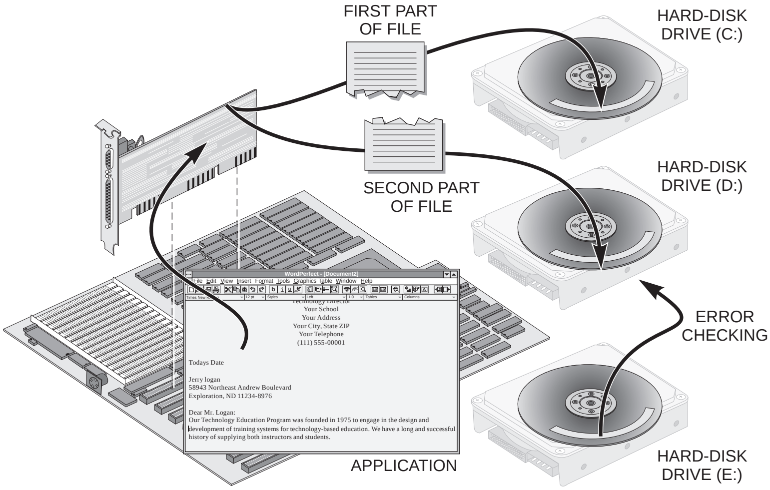


Duplicate information is stored on both drives. When the file is retrieved from the array, the controller reads alternate sectors from each drive. This effectively reduces the data read time by half.

The `RAID 2` strategy interleaves data on parallel drives, as described in Figure 8.28. Bits or blocks of data are interleaved across the disks in the array. The speed afforded by collecting data from the disks in a parallel format is the biggest feature of the system. In large arrays, complete bytes, words, or double-words can be written to, and read from, the array simultaneously.

Figure 8.28

Interleaved data on parallel drives.



The RAID 2 specification uses multiple disks for error-detection and correction functions. Depending on the error-detection and correction algorithms used, large portions of the array are used for non-data storage overhead. Of course, the reliability of the data being delivered to the system is excellent, and there is no need for time-consuming corrective read operations when an error is detected. Arrays dealing with large systems may use between three and seven drives for error-correction purposes. Because of the high hardware overhead, RAID 2 systems are not normally used with microcomputer systems.

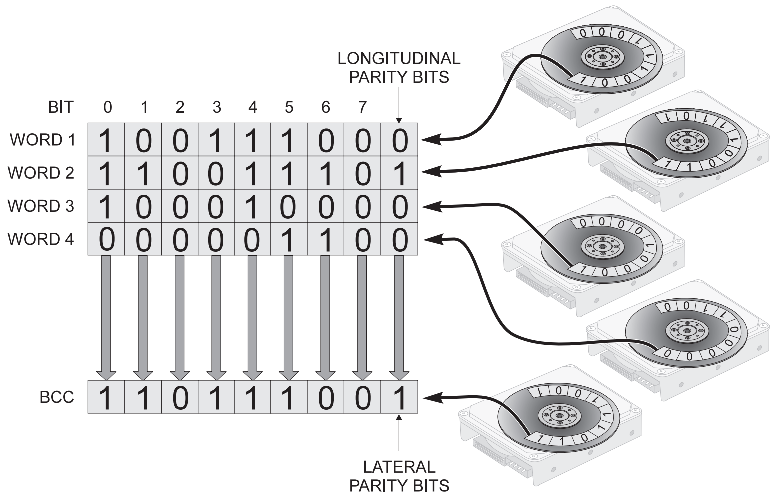
When the array is used in this manner, a complex error-detection and correction algorithm is normally employed. One of the easiest algorithms to understand involves generating a parity bit for each word applied to the array, and then grouping the data words into blocks. Each block is then used to generate a parity bit. In this manner, a single error will cause a failure in both the word's parity bit and one of the block's parity bit. The intersection of the two failed parity bits marks the location of the erroneous data bit. Because the system is binary, the only action required to correct the bit is to change it to the other possible value (`0=1` or `1=0`). The controller contains circuitry that detects, locates, and corrects the error without retransmitting any data. This is a very quick and efficient method of error-detection and correction. For more critical application, extravagant mathematical formulas may be employed for the error-correction algorithm to ensure the integrity of important data.

In Figure 8.27, the data block being sent to the array is broken apart and distributed to the drives in the array. The data word already has a parity bit added to it. The controller generates parity for the block and stores it on the error-detection drive. When the controller reads the data back from the array, it regenerates the error-check character and compares it to the one written on the error-check drive. By comparing the error-check character to the rewritten one, the controller can detect the error in the data field and determine which bit within the field is incorrect. With this information in hand, the controller can just correct that bit as it is being processed.

When the data is broken into blocks, as described for RAID 2, the parity bit generated for the whole word is referred to as a `longitudinal parity bit`. The parity bits generated for the block are called `lateral parity bits`. The lateral parity bits are assembled into the `block-check character` (BCC) that is stored on the error-detection drive. The block-check character is used in a type of error correction known as `longitudinal redundancy checking`. The more advanced mathematical models for error correction are known as `cyclic redundancy checking` (CRC). The concept of lateral and longitudinal parity is described in Figure 8.29.

Figure 8.29

Lateral and longitudinal parity.



In a `RAID 3` arrangement, the drives of the array operate in parallel like a RAID 2 system. However, only parity checking is used for error-detection and correction—requiring only one additional drive. If an error occurs, the controller reads the array again to verify the error. This is a time-consuming, low-efficiency method of error correction.

A `RAID 4` controller interleaves sectors across the drives in the array. This creates the appearance of one, very large drive. The RAID 4 format is generally used for smaller drive arrays, but can be used for larger arrays as well. Only one parity-checking drive is allotted for error control. The information on the parity drive is updated after reading the data drives. This creates an extra write activity for each data read operation performed.

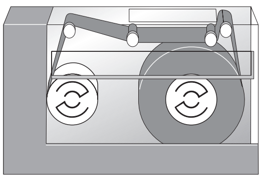
The `RAID 5` scheme alters the RAID 4 specification by allowing the parity function to rotate through the different drives. Under this system, error-checking and correction is the function of all the drives. This is usually the most popular RAID system, because it can be used on arrays as small as two drives, with a high level of error recovery built in.

Tape Drives

Tape drive units are another popular type of information storage system. These systems can store large amounts of data on small `tape cartridges`, similar to the one depicted in Figure 8.30. Tapes tend to be more economic that other magnetic media when storing large amounts of data. However, access to information stored on tape tends to be very slow. This is caused by the fact that, unlike disks, tape operates in a linear fashion. The tape transport must run all the tape past the drive's R/W heads to access data physically stored at the end of the tape.

Figure 8.30

Data storage on small tape cartridges.



Therefore tape drives are generally used to store large amounts of information that will not need to be accessed often, or quickly. Such applications include making backup copies of programs and data. This type of data security is a necessity with records such as business transactions, payroll, artwork, and so forth.

Data backup has easily become the most widely used tape application. With the large amounts of information that can be stored on a hard-disk drive, a disk crash is a very serious problem. If the drive crashes, all the information stored on the disk can be destroyed. This can easily add up to billions of pieces of information. Therefore an inexpensive method of storing data away from the hard drive is desirable.

Early personal computers used audio cassette tapes as storage units for data. The original PCs had an adapter for connecting a cassette recorder to it. When hard- and floppy-disk drives became popular, tape drives began to disappear. As the size of hard drives began to make backup on floppy disks inconvenient, however, tape systems began to come back into acceptance.

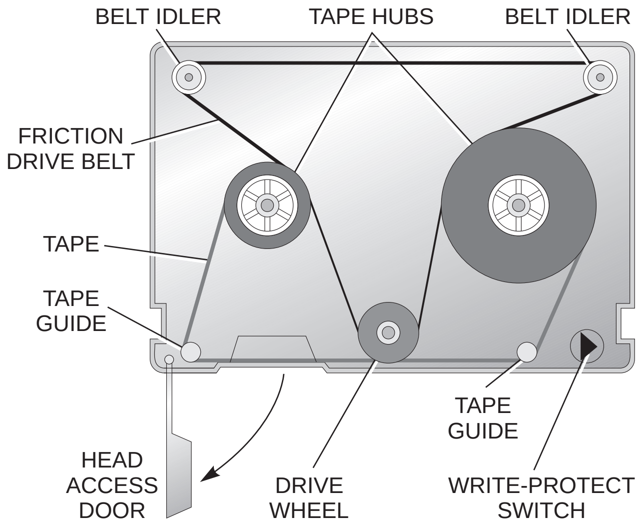
Tape Standards

As more users employed tape as a backup media, standards for tape systems were formed. The most widely used tape standard is the `quarter-inch cartridge` (QIC) standard. This standard calls for a tape cartridge like the one depicted in Figure 8.31. Its physical dimensions are 6X4X5/8 inches. The cartridge has a head access door in the front that swings open when it is inserted in the drive unit.

Unlike the audio cassette tape, the cartridge tape is not driven by capstans that extend through the tape spools. Instead, it employs a belt drive system that loops through the cartridge and turns both spools synchronously. The belt is driven by a rubber drive wheel, which in turn is driven by the capstan. This design provides smoother, more precise operation than the audio cassette is capable of.

Figure 8.31

A 1/4-inch tape cartridge.



The R/W heads magnetize the tape as it passes by, in much the same manner as described for other magnetic storage media. The data is placed on the tape serially as it moves past the head. The tape is organized into sectors of data, separated by intergap blocks. The tracks of data can be applied in parallel (using multiple R/W heads) in a continuous stream of data (`streaming tape systems`) or in a `serpentine` manner, where the data is applied to the tape in one direction for odd tracks, and in the other direction for even tracks.

Magnetic tape must be formatted before use, just like a disk. In the formatting process, the controller marks the tape off into sectors. In addition, it establishes a file allocation table in its header, similar to that of a floppy or hard disk. The header also contains a bad-sector table to prevent defective areas of the tape from being used. Some of the tape is devoted to the error-detection and correction information that must be used with tape systems. Tape systems use the cyclic redundancy checking and other mathematical algorithm methods described in the discussion of RAID systems.

Cartridge tapes were referred to as `DC-6000` style tapes. Their model numbers would normally include a reference to their tape capacity as the last digits (that is, DC-6200 would be a 200 MB tape). As the cartridge tape industry matured, manufacturers came together to establish standards for tape formats and labeling. In the process, the DC-6000 number has been replaced in discussions about capacity and format.

For the most part, a series of QIC numbers have been used to describe different tape cartridges. Table 8.4 provides a sample list of QIC standard numbers.

Table 8.4

QIC specifications.

Specification	Tracks	Capacity	Cartridge
QIC-02	9	60 MB	DC-3000
QIC-24	9	60 MB	DC-6000
QIC-40	20	40 MB	DC-2000
QIC-80	32	80 MB	DC-2000
QIC-100	12/24	100 MB	DC-2000
QIC-150	18	250 MB	DC-6000
QIC-1000	30	1.0 GB	DC-6000
QIC-1350	30	1.35 GB	DC-6000
QIC-2100	30	2.1 GB	DC-6000

A minicartridge version of the quarter-inch tape cartridge, with dimensions of 3 1/4X2 1/2X5/8 inches, has been developed to provide a more compact form factor to fit in 3 1/2-inch drive bays. The internal operation of the cartridge has remained the same, but the amount of tape inside has been reduced. The reduced amount of tape in the cartridge is offset by the use of more advanced data-encoding schemes that store more data on less tape. Minicartridges are referred to as DC-2000-style cartridges. Like the DC-6000 tapes, the DC-2000 model numbers will normally include a reference to their tape capacity as the last digits.

A number of QIC tape standards have developed over time. The original QIC standard was `QIC-40.` This standard called for a unit that could be connected to a floppy-disk drive interface so that it acted like a large B: drive. It specified a 20-track format, with each track consisting of 68 segments, having 29 sectors of 1,024 bytes each. This format provided 40 MB of data storage. The specification treated the tape's sectors like the sectors of a floppy disk, in that they were organized into files.

An updated `QIC-80` specification was developed to replace the QIC-40 standard. Advanced R/W head structures allow the QIC-80 to place 32 tracks on the tape rather than 20. Coupled together with improved data-per-inch storage capabilities, the total capacity of the cartridge was boosted to 80 MB. The QIC-80 systems included data compression software that could effectively double the capacity of the drive from its stated value.

The QIC-80 standard has been superseded by the `QIC-500M` format, that allows for up to 500 MB of data to be stored on the cartridge. Newer standards for tape drives continue to emerge. Specifications that depart from the floppy-disk drive interface, and use the IDE or SCSI interfaces, are producing data storage potentials into the multiple gigabyte ranges.

Tape Drive Troubleshooting

Because the fundamentals of recording on tape are so similar to those used with magnetic disks, the troubleshooting process is also very similar. The basic components associated with the tape drive include the tape drive, the signal cable, the power connection, the controller, and the tape drive's operating software.

The tape itself can be a source of several problems. Common points to check with the tape include the following:

- Is the tape formatted correctly for use with the drive in question?
- Is the tape inserted securely in the drive?
- Is the tape write-protected?
- Is the tape broken or off the reel in the cartridge?

As cartridge tapes are pulled back and forth, their Mylar base can become stretched over time. This action can cause the tape's format to fail before the tape actually wears out. To remedy this, the tape should be retentioned periodically using the software's retention utility. Cartridge tapes are typically good for about 150 hours of operation. If the number of tape errors begins to increase dramatically before this time, try reformatting the tape to restore its integrity. After the 150 hour point, the tape should just be replaced.

If the tape is physically okay and properly formatted, the next easiest section to check is the tape software. Check the software Setup and Configuration settings to make sure they are correct for any hardware settings. Refer to the tape drive's User's Guide for a list of system requirements, and check the system to make sure they are being met.

If any configuration jumpers or switches are present on the controller, verify that they are set correctly for the installation. Also, run a diagnostic program to check for resource conflicts, such as IRQ and base memory addressing, that may be preventing the drive from operating.

The software provided with most tape drive units include some error messaging capabilities. Observe the system and note any tape-related error messages it produces. Consult the User's Manual for error message definitions and corrective suggestions. Check for error logs that the software may keep. These logs can be viewed to determine which errors have been occurring in the system.

Because many tape drive's are used in networked and multi-user environments, another problem occurs when you are not properly logged on or enabled to work with files being backed up or restored. In these situations, the operating system may not allow the tape drive to access secured files, or any files, because the correct clearances have not been met. The network administrator should be consulted for proper password and security clearances. See Chapter 11, "Data Communications," for more information about the network environment.

Reinstall the drive's software and reconfigure it. Go through the installation process carefully, paying close attention to any user-selected variables and configuration information requested by the program.

If hardware problems are suspected, begin by cleaning the drive's R/W heads. Consult the User's Guide for cleaning instructions, or use the process described in Chapter 13 for manual cleaning floppy-disk drive R/W heads. The R/W heads should be cleaned after about 20 backups or restores. Also, try using a different tape to see whether it works. Make certain that it is properly formatted for operation. It should also be a clean tape if possible, to avoid exposing any critical information to possible corruption.

If cleaning does not restore the drive to proper operation, continue by checking the power and signal cables for good connection and proper orientation.

Because matching tape drives are not common at a single location, checking the drive by substitution should be considered as a last step. Check the User's Guide for any additional testing information and call the drive manufacturer's technical-service number for assistance before replacing the drive.

Summary

This chapter has explored the use of magnetic media to provide long-term, high-volume data storage for personal computer systems. The first section of the chapter explained the fundamentals of recording and reading data with magnetic media. This introduction was followed by an extensive explanation of floppy-disk drive operations and troubleshooting. The third section dealt with installation, operation, and troubleshooting of hard-disk drive systems. The final sections of the chapter covered other magnetic storage devices and strategies commonly used with personal computer systems.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand and can perform each item listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercise

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 26 - FDD Problem Isolation, 27 - HDD Installation, and 28 - HDD Problem Isolation.

Review Questions

1. The maximum size of a disk partition using DOS 3.x is ______.
2. What action should always be taken before upgrading a hard-disk drive?
3. Head-to-disk interference (HDI) is also referred to as ________.
4. What corrective steps should be taken if the system will not boot up to the C:\> prompt, but will allow it to be accessed after booting from a floppy boot disk?
5. Which interrupt request channel is normally used with floppy-disk drives in a PC-compatible system?
6. List the types of HDD interfaces that are system-level interfaces. What does this designation imply?
7. How can the A: and B: floppy-disk drives be differentiated by looking into the system unit?
8. What action should be taken if a `Disk Boot Failure` message is displayed on the monitor screen?
9. List the HDD-related hardware components tested by the chapter's HDD troubleshooting procedure, in the order they were checked.
10. If the system will not boot up to the hard drive, but can be accessed after booting to a floppy-disk drive, what type of problem is indicated?

11. List the nonhardware items checked by the HDD troubleshooting procedure, in the order they were checked.
12. What precaution should always be taken before reformatting a hard-disk drive? Why?
13. List five conditions that could cause a `Bad or missing COMMAND.COM` message to be displayed onscreen.
14. In which types of applications should a tape drive backup system be considered?
15. Describe how data is stored on a magnetic disk.
16. List the steps involved in installing a hard-disk drive in a desktop system.
17. A group of logically related disk sectors is called ______.
18. Describe the differences between the two types of drive array applications.
19. How is a cartridge tape different than a standard audio cassette tape, and why is it better?
20. What is the main purpose of a RAID 1 drive array?
21. What is formatting as it applies to a disk?
22. What function does the DOS FDISK program perform?
23. What type of interface does a common tape drive use?
24. What is the major procedural difference between installing a floppy-disk drive and a hard-disk drive?
25. What is the main function of the DOS high-level format?

Review Answers

1. 32 MB. For more information, see the section titled "Logical and Physical Drives."

2. The contents of the drive should be backed up to some other media before exchanging or upgrading a hard drive unit. For more information, see the section titled "HDD Installation."

3. A head crash. For more information, see the section titled "Contact versus Noncontact Recording."

4. The system files should be transferred from a system disk back to the root directory of the HDD. For more information, see the section titled "Software Checks."

5. IRQ-6. For more information, see the section titled "FDD Controller."

6. IDE and SCSI. The interfaces for these drive types are located on the drive's themselves. Therefore the drive only communicates with the system, not an adapter card. For more information, see the section titled "HDD Interfaces."

7. The A: drive is located at the end of the signal cable; the B: drive is always connected to the middle of the signal cable. A twist of wires in the cable between the A: and B: connectors creates the difference between the two connections. For more information, see the section titled "Floppy-Drive Cables."

8. Try to boot the system to the A: drive and check to see whether the hard drive can be accessed. If so, attempt to replace the system boot files on the hard drive. For more information, see the section titled "Software Checks."

9. The HDD controller card (pre-Pentium), the signal cable(s), and the HDD unit. For more information, see the section titled "HDD Hardware Checks."

10. Missing system files on the HDD. For more information, see the section titled "Software Checks."

11. The drive type settings and parameters in the CMOS setup and the system files in the root directory. For more information, see the section titled "HDD Configuration Checks and Software Checks."

12. Before reformatting a drive, use the DOS `VER` command to determine which version of DOS is currently in use. For more information, see the section titled "Logical and Physical Drives."

13. (1) No bootable files on the hard drive and no bootable disk in the FDD, (2) No COMMAND.COM file located in the HDD's root directory, (3) when installing a new hard drive unit, (4) when the COMMAND.COM file has been erased from the root directory, and (5) when a new copy of DOS has been installed. For more information, see the section titled "Software Checks."

14. Any application where large amounts of information need to be stored in a nonvolatile manner. For more information, see the section titled "Tape Drives."

15. Data is stored on the disk in the form of positively and negatively charged spots. The spots are encoded to represent bits. The bits are formed into bytes that are stored in sectors along the tracks of the disk. For more information, see the section titled "Reading and Writing On Magnetic Surfaces."

16. Obtain the drive's type information, check its single/master/slave/termination settings, slide it into a drive bay, insert the retaining screws in the drive, connect the signal cable(s) between the drive and the controller, connect the HDD power connector, start the system, enter the CMOS Setup screen, load the HDD parameters, and format/partition the drive. For more information, see the section titled "HDD Installation."

17. A cluster. For more information, see the section titled "Disk Drive Operations."

18. The first drive array type is used as a data backup method. The second type of drive array uses multiple drives to store large amounts of data in a manner that provides high reliability and recoverability. For more information, see the section titled "RAID Systems."

19. The cartridge tape is driven by a belt rather than two capstans. This provides smoother operation than possible with an audio tape. For more information, see the section titled “Tape Standards.”

20. As a data backup method. For more information, see the section titled “RAID Systems.”

21. Preparing the disk to hold data. For more information, see the section titled “Magnetic Disks.”

22. It establishes the logical drive boundaries, called partitions, on the physical disk. For more information, see the section titled “Logical and Physical Drives.”

23. Older units used the system’s FDD controller and acted as the B: drive. Newer units use proprietary, IDE, and SCSI interfaces. For more information, see the section titled “Tape Standards.”

24. The HDD’s type and operating parameter information must be installed in the CMOS setup. The HDD unit must be partitioned and formatted before it can be used. For more information, see sections titled “Installation” under “Floppy-Disk Drives,” and “HDD Installation.”

25. To establish the placement of tracks and sectors on the drive and create the FAT to locate them. For more information, see the section titled “Formatting.”

Chapter 9

Video Displays

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- List the components commonly found inside a CRT monitor.
- Describe the physical aspects of a cathode ray tube.
- Using raster scanning, explain how a single dot can be positioned anywhere on the face of the CRT.
- Describe how a character generator is used to convert ASCII-coded characters into screen images.
- Differentiate between composite and RGB video signals.
- Describe how color displays are created onscreen.
- Define bitmapped graphics.
- Define the terms `pixel` and `PEL`.
- Describe the function of a shadow mask in a CRT monitor.
- State the characteristics of the MGA, CGA, HGA, EGA, and VGA video standards.
- Describe the type of physical connector specified for each video standard.

continues

- Explain how the amount of installed video memory affects the capabilities of a video adapter card.
- Discuss safety considerations associated with working around a CRT.
- Describe the operation of gas-plasma and liquid crystal displays.
- Describe steps to troubleshoot monitor problems.
- Describe steps to troubleshoot video problems.
- Identify symptoms associated with common monitor problems.
- Describe how video adapters typically store attribute information in video memory.
- Apply FRU troubleshooting steps to exchange components in CRT monitors.

Introduction

The video monitor has long been one of the most popular methods of displaying computer data. At the heart of the monitor is the cathode ray tube (CRT), familiar to us from the television receivers we have in our homes. As a matter of fact, the early personal computers used televisions as video units. The basic difference between the television and a monitor is that no radio-frequency demodulation electronics are used in the video monitor.

As an output device, the monitor can be used to display alphanumeric characters and graphic images. There are two possible methods used to create these displays: the raster-scan method, and the X-Y, or vector, scan method. All television sets, and most video displays, are of the raster-scan type. Therefore, this is the type we will concentrate on in this text. An oscilloscope display is a prime example of vector scanning.

The popularity of portable computers has created a large market for lighter display devices. The main devices used in this market are the LCD and gas-plasma displays. These devices do not use a CRT tube or its supporting circuitry. Therefore the weight associated with the CRT and its high-voltage components is not present. The flat-panel nature of these devices also works well in the portable computer because of its reduced size.

CRT Basics

Objective

Quite simply, a CRT is an evacuated glass tube with an electron gun in its neck and a fluorescent-coated surface opposite the electron gun. Figure 9.1 shows a typical CRT. When activated, the electron gun emits a stream of electrons that strike the fluorescent coating on the inside of the screen, causing an illuminated dot to be produced.

The position of the beam along the face of the screen can be manipulated through horizontal and vertical deflection coils attached to the tube. They cause the beam to deflect, according to electromagnetic attraction and repulsion principles. The deflection coils are usually combined into a single unit, called a yoke, that slips over the neck of the tube. By applying specific signals to the coils, the beam can be positioned anywhere along the face of the screen. In the raster-scan method of creating displays, separate signals are applied to the horizontal and vertical deflection coils to cause the electron beam to move across the screen. As the beam moves, it leaves an illuminated trace that requires a given amount of time to dissipate. The amount of time depends on the characteristics of the fluorescent coating. This dissipation quality is referred to as `persistence`.

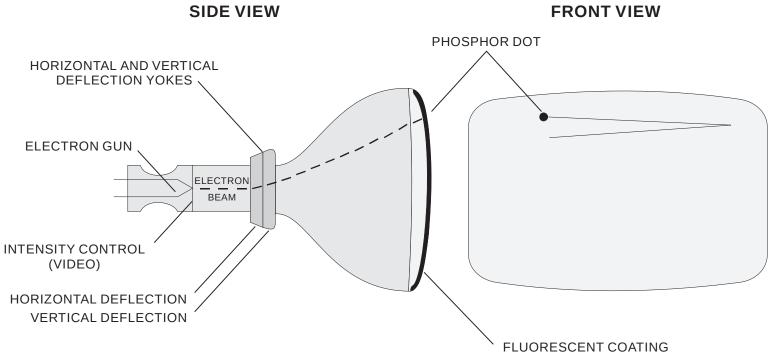


Figure 9.1

A cathode ray tube.

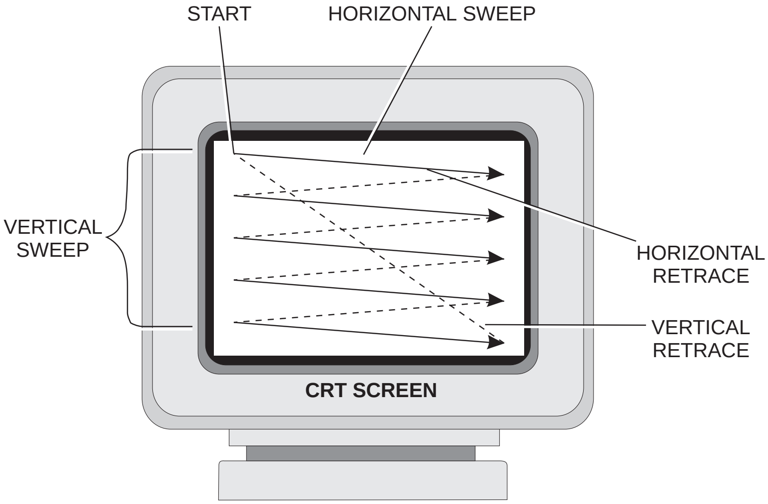
In theory, the electron beam begins at the upper-left corner of the screen and sweeps across its face to the upper-right corner, leaving a line across the screen. This is called a `raster line`. After reaching the right side of the screen, the trace is blanked-out and the electron beam is repositioned to the left side of the screen, one line below from the first trace (`horizontal retrace`). At this point, the `horizontal sweep` begins producing the second display line on-screen. The scanning continues until the horizontal sweeps reach the bottom of the screen, as shown in Figure 9.2. At that point, the electron beam is blanked again and returned to the upper-left corner of the screen (`vertical retrace`), completing one `field`.

Creating Video Information

Video information is introduced to the picture by varying the voltage applied to the electron gun as it scans the screen. Typically, a voltage of just above 1 V applied to the electron gun's drive circuitry produces `no` electron emission, and a black (blank) area is created. A signal voltage of approximately 3 V causes maximum electron emission, and a "white" area is created. Voltages between 1 and 3 volts result in various levels of gray. In this manner, the electron gun paints the desired picture on the fluorescent screen by varying its intensity.

Figure 9.2

Raster-scan video.



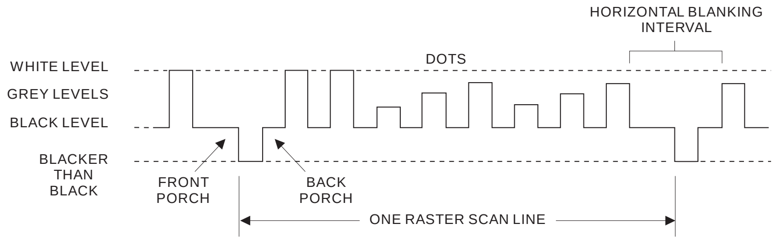
The human eye perceives only the "picture," because of the blanking of the retrace lines and the frequency at which the entire process is performed. Typically, a horizontal sweep requires about 63 microseconds to complete; a complete field requires approximately 1/60 of a second, or 1/30 of a second per frame. The National Television Standards Committee (NTSC) specifies 525 lines per frame, composed of two fields of 262.5 lines, for television pictures. The two fields—one containing the even-numbered lines, and the other containing the odd-numbered lines—are interlaced to produce smooth, flickerless images. This method of creating display images is referred to as `interlaced scanning`, and is primarily used with television. Most computer monitors use a `non-interlaced scanning` method, as described in Figure 9.2. In non-interlaced scanning, the entire image is painted onscreen before a vertical retrace is performed.

CRT Signals

Figure 9.3 depicts a sample voltage signal delivered to the CRT for one raster of a field, containing white and various shades of gray dots on a black background. At the end of each raster scan, a `horizontal synchronization` (HSYNC) signal is applied to the electron gun to cover the retrace. The area designated as the "back porch" produces a black frame at the right edge. A `vertical synchronization` (VSYNC) signal is introduced at the bottom of the field to cover the movement of the beam back to the top of the screen.

Figure 9.3

A single raster-scan signal.



The `Cathode Ray Tube Controller` (CRTC) develops the video signals and the horizontal and vertical synchronization signals for the CRT. The HSYNC and VSYNC signals are applied to free-running horizontal and vertical sweep oscillators, to allow the video information to be synchronized with the scanning motion of the electron beam. Other monitors accept a single input signal composed of all three signals, referred to as a `composite video signal.`

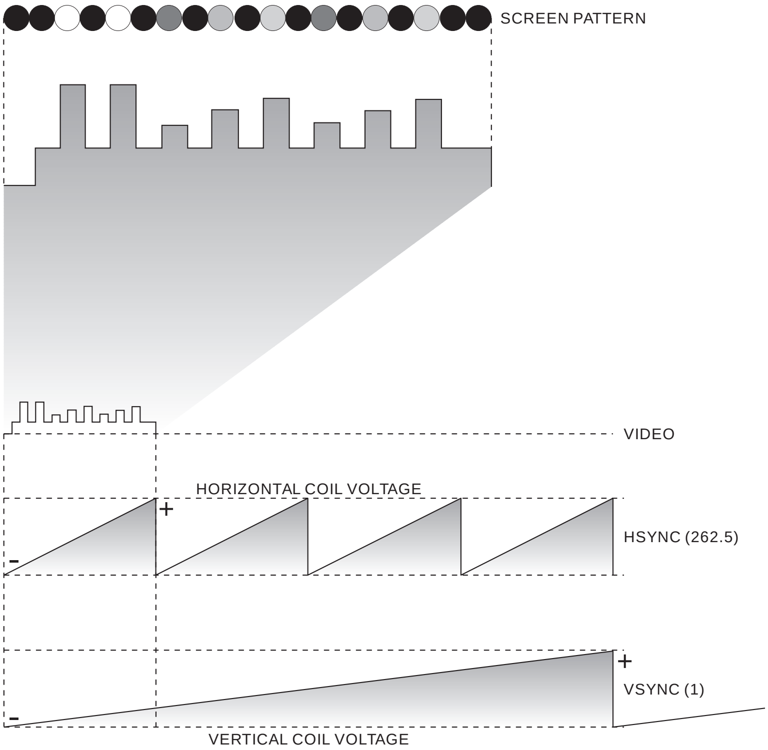
Figure 9.4 illustrates the relationship between the HSYNC, VSYNC, and video signals. As the electron beam starts in the upper-left corner of the screen, current through the horizontal and vertical coils is at minimum. To begin the scanning process, current is applied to the upper-vertical and left-horizontal coils, creating a positive polarity field.

At the same time, the current passing through the lower-vertical and right-horizontal coils sets up a negative field. The polarities of these fields deflect the negatively charged stream of electrons to the right and down the screen, as noted in Figure 9.4. The sawtooth waveforms applied to the coils control the positioning of the beam onscreen. As the horizontal waveform becomes more positive, the beam is deflected further across the screen to the right. Likewise, the more positive the vertical waveform, the further the beam is pushed down the screen.

In most monitors, the frequency relationships between these signals is fixed. However, there are some monitors, referred to as *Multisync monitors*, that can adapt to various horizontal sweep rates and vertical refresh rates to accommodate a variety of different video standards. Some of these standards are discussed later in this chapter (see Figure 9.20).

Figure 9.4

HSYNC, VSYNC, and video signal relationships.



A serial line of dot information is applied to the electron gun as the horizontal coil waveform deflects the beam across the viewing area. The data stream is delayed at the beginning and end of each scan line so that the data falls within the boundaries of the screen. The `front` and `back porches` of the video signal create a blank border at the right and left sides of the screen. The horizontal sync pulse produces a `blacker-than-black` situation that covers the horizontal retrace of the electron gun.

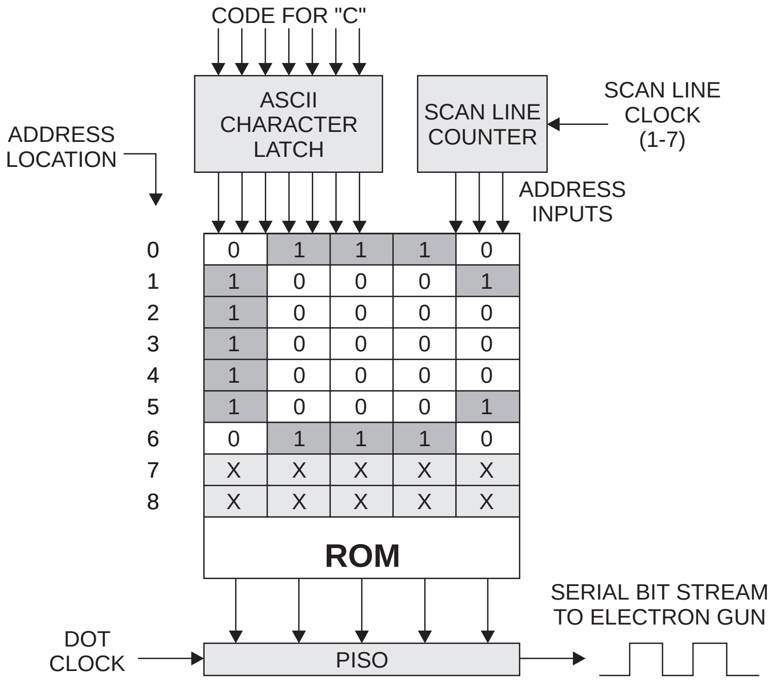
Video Character Generators

So far, we have shown how dots could be displayed on a CRT. The next step is to examine the intelligence used to create meaningful images onscreen. The most widely used monitor application is to display alphanumeric characters onscreen. The dots hold the key to this task. A component of the CRTC, called the `character generator`, converts the ASCII code for each character into a 5×7 or 7×9 dot-matrix character. A number of other character sizes can also be used.

The character generator is basically a ROM device containing dot-pattern information for the entire character set. Because only two intensity levels are required (white and black) to display alphanumeric data, simple digital logic levels can be used to produce the two levels. Figure 9.5 shows one section of a 5×7 dot-matrix character generator ROM, and some of the circuitry necessary to convert the character code into a pulse train that can be used to drive the electron gun.

Figure 9.5

Character generation.



Notice that the dot pattern for the letter `C` is stored in seven consecutive addresses. The ASCII-coded `C` is applied to the ROM as part of its address inputs. The other part of the address input is derived from a counter that counts 0 through 6. The ASCII code accesses the block of memory containing the `C`, while the counter selects the first address in the block with its initial count state. The contents of the first address is applied to a parallel-in, serial-out shift register, which produces the first section of the video pulse train. When the five bits have been shifted out of the register, the line counter is pulsed. With this, the second ROM location is accessed and shifted out. This sequence of events continues until all seven locations have been accessed and serialized. The line counter pulse is synchronized with the HSYNC pulses so that the `C` is reconstructed on the CRT over seven successive horizontal sweeps.

This example contains all the circuitry necessary if only one character is to be displayed on each `character line` of the CRT. Of course, several characters are normally displayed on a character line. A common monitor arrangement calls for 25 lines of text, with 80 characters per line. This requires for the first address of up to 80 character blocks to be serialized during the first horizontal trace of the CRT.

By loading the character codes into an 80-character line buffer memory, and then sequentially addressing it (while holding the line counter), the characters in the buffer are accessed in order and applied to the character address latch. This produces the first row of each character's dot pattern.

After the first row of 80 characters has been serialized, the line counter is pulsed and the sequential addressing of the line buffer is repeated. This produces the second row of each character during the second horizontal trace of the CRT. Therefore the line buffer must be accessed through seven cycles to produce the complete line of text on the CRT, as illustrated in Figure 9.6.

After a line, or a page, of text has been displayed onscreen, it must be rewritten periodically to prevent it from fading away. For the rewrite to be performed fast enough to avoid display flicker, the contents of the display are stored in a special memory, called the `screen memory`. The capacity of the memory depends on the characteristics of the display. In the example of 25 lines of text at 80 characters per line, the screen memory must be able to hold 2,000 bytes of ASCII character information (80×25=2,000 characters).

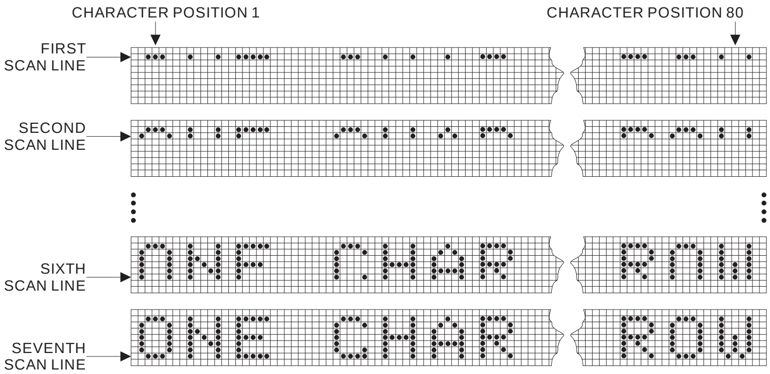


Figure 9.6

A video text line.

Storing Text Information

Objective

The 80×25 format is typical for alphanumeric text mode in most monitor types. When the adapter is in text mode, it will typically require at least two bytes of screen memory for each character position onscreen. As you have seen, the first byte is for the ASCII code of the character itself. The second byte is used to specify the `screen attributes` of the character and its cell. Under this condition, the screen memory in the example must hold at least 4000 bytes. The `attribute byte` specifies how the character is to be displayed. Common attributes include underlining, blinking, and the color of a text character for the color displays.

The character box is divided into two parts, the foreground and the background. The foreground consists of the dots that make up the character itself. The background is made up of the other dots in the character cell. The attribute byte can control the intensity of the foreground or background. For monochrome displays, there are only four combinations of the foreground and background (excluding the highlight and blink).

For color displays, the attribute byte can be used to specify different colors for the foreground and background. The highlight bit is combined with the foreground bits to provide up to 16 different colors for the foreground. Using three bits to define the background, it is possible to have eight different colors for the background.

CRT Controllers

Although a CRT controller may consist of a simple ROM character generator and a few logic devices, the CRT controller arrangement is so common that IC manufacturers produce it in fully integrated packages. In addition to the functions previously stated, the CRTC must perform a number of other tasks. In general, the CRTC is responsible for generating the dot-row, HSYNC, and VSYNC signals to synchronize the timing of the video signals and for refreshing the display. It also controls display-manipulation functions such as scrolling, paging, inverse video, character brightness, and cursor positioning (discussed later in this chapter).

Because the CRTC must continually access the screen memory to refresh the screen, there is a natural `contention` between the CRTC and the system microprocessor, which must also access the screen memory to enter new data. This contention may be resolved in a number of ways. The microprocessor access times can be limited to certain periods, such as horizontal and vertical retrace, when the CRTC does not need to access the screen memory. In systems that use a two-phase system clock, microprocessors use the system buses during one phase only. Therefore, the CRTC can access the buses on the other phase. An external DMA controller can also be used to multiplex the transfer of information between the microprocessor, or CRTC, and the screen memory. Each of these concepts requires some additional logic circuitry to determine access rights to the screen memory.

Color Monitors and Graphics

Objective

The monitor discussed so far is referred to as a `monochrome` monitor because it is capable of displaying only shades of a single phosphor color. A color monitor, on the other hand, differs from a monochrome monitor. The major variance lies in the construction of color CRTs.

The basic construction of a color CRT is shown in Figure 9.7. It uses a combination of three-color phosphors—red, blue, and green—arranged in adjacent trios of dots or bars, called `picture` elements, pixels, or PELS. By using a different electron gun for each element of the trio, the individual elements can be made to glow at different levels to produce almost any color desired. The electron guns scan the front of a screen in unison, in the same fashion described earlier for a monochrome CRT. Color CRTs add a metal grid in front of the phosphor coating called a shadow mask. It ensures that an electron gun assigned to one color doesn't strike a dot of another color.

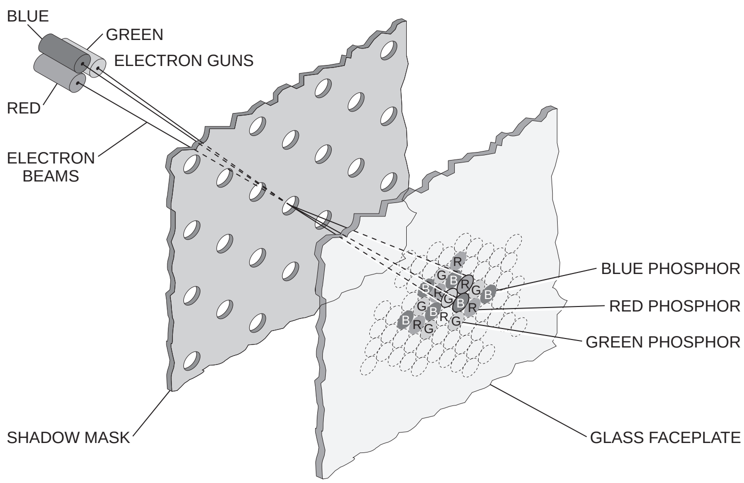


Figure 9.7

Color CRT construction.

The quality of the image produced onscreen is a function of two factors: the speed at which the image is retraced onscreen, and the number of pixels onscreen. The more pixels on a given screen size, the higher the image quality. This quantity is called resolution, and is often expressed in an X-by-Y format. The X portion of the specification refers to the number of horizontal dots the monitor can display; the Y function is the number of vertical dots possible. Using this format, the quality of the image is still determined by how big the viewing area is (that is, an 800×600 resolution on a 14-inch monitor will produce much better quality than the same number of pixels spread across a 27-inch monitor).

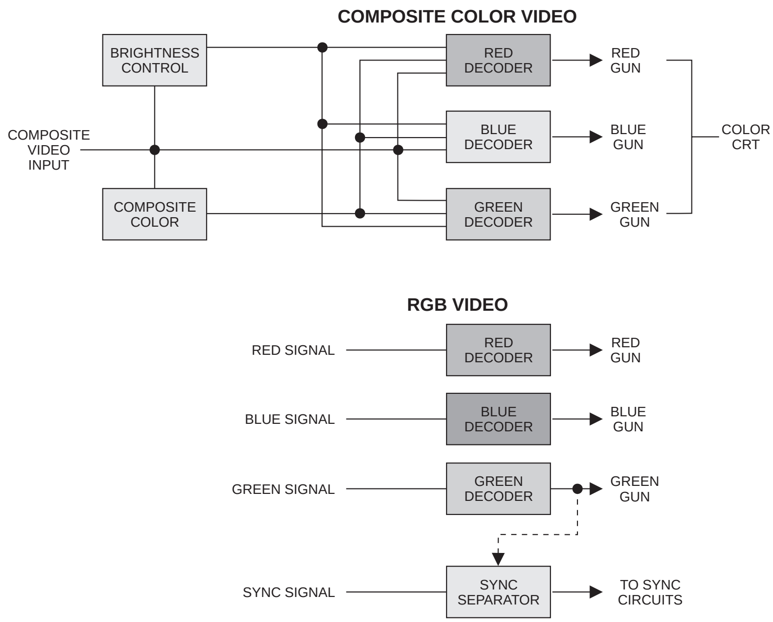
Resolution can be expressed as a function of how closely pixels can be grouped together onscreen. This form of resolution is expressed in terms of dot pitch. A monitor with a .28 dot pitch has pixels that are located .28 mm apart. In monochrome monitors, dot pitch is measured from center to center of each pixel. In a color monitor, the pitch is measured from the center of one dot trio to the center of the next trio.

Types of Color Monitors

Color monitors fall into two basic groups, determined by the type of video signal they use. These are `composite color monitors` and `RGB monitors`. Composite monitors combine all the color information, synchronizing information, and brightness into a single signal that must be decoded by the monitor. This type of signal processing is illustrated in Figure 9.8. The RGB (red-green-blue) color monitor uses separate signals for each color and sync signal, and thus offers better color control than composite monitors. The RGB method of delivering color signals to the monitor is also depicted in the figure. Typical composite monitors offer resolutions of about 260×300 pixels. High-resolution RGB monitors offer resolutions above 1024×768 pixels.

Figure 9.8

Composite and RGB video drives.



Actually, there are two subtypes of RGB monitors. The first is a `TTL-compatible` monitor, capable of generating a limited number of colors (eight). Information presented to the three electron guns is digital in nature; therefore each gun can only be "on" or "off" for a given phosphor trio. This limits the number of colors that the monitor can generate. The other RGB monitor type accepts `analog` input levels that can independently vary the intensity of each gun to produce a nearly endless range of colors.

For displaying alphanumeric text material (letters and numbers), the color monitor has no significant advantage over the monochrome monitor. The monochrome monitor typically offers high resolution, at a much lower cost, than color monitors. This is due to the extra complexity of the color monitor, which requires expanded memory capacity, an extra decoder, and amplifier circuitry to process the color signals.

Color monitors have an edge over monochrome monitors when there is something to be gained by the addition of color to the display. These applications are generally in the area of `graphics display` (pictorial representations). The average composite monitor possesses sufficient color control and resolution to handle simple color graphics associated with home video games and elementary graphics, such as bar and pie charts. Advanced high-resolution color graphics, however, require the high resolution and color control offered by analog RGB monitors.

CRT Graphics

Perhaps the simplest form of graphics is contained in the CRTC's character generator. In addition to the ASCII character set, character generators also hold a set of special graphic shapes called `block-graphics`. These shapes depicted in Figure 9.9 can be joined together to generate lines, curves, and other graphic representations, including basic line diagrams.

Another simple approach to graphics involves turning all the dots in a character block on or off. In this manner, solid blocks of dark and light spaces are created, which can be used to form rough geometric shapes. Applying this concept to a 25-line-by-80-character display, 2,000 picture elements are created that can be turned on or off independently. A pixel is defined as the smallest block in a graphic display that may be turned on or off independently. To improve the quality and resolution of the graphics, character blocks can be subdivided into smaller, on-off addressable blocks. By creating more and smaller pixels, the rough edges of the display are smoothed out and the resolution of the display is increased in proportion to the number of subdivisions. If the display's character blocks are divided into six sub-blocks, the number of pixels is increased to 12,000, and the picture resolution is six times greater.

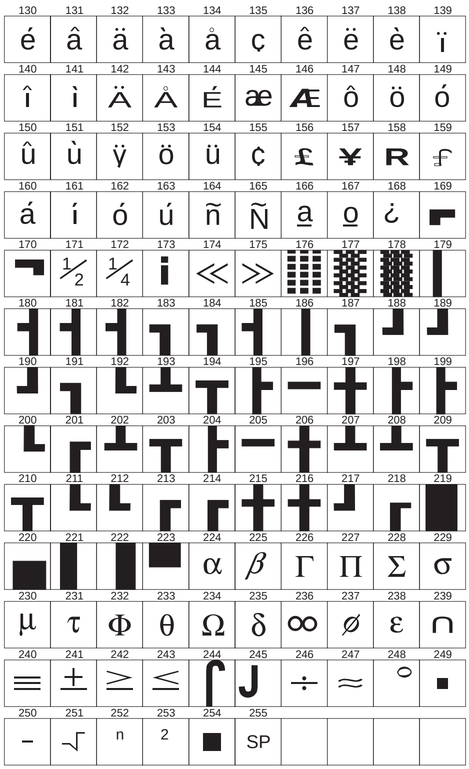


Figure 9.9

Special graphics shapes.

The ultimate end to subdividing the display occurs when each dot onscreen is mapped to a specific memory data bit. This approach to creating graphics is referred to as `bitmapped graphics`. Of course, increasing the number of pixels in the display means more data must be stored in memory. Instead of storing 80-character addresses per character-line and then repeating them to generate the line, as we did with alphanumeric characters, the system must store one bit for each pixel. If color or analog data is also used, the display system must be supported by an extensive RAM memory. For this reason, many systems may offer more than one bit-mapped graphic format, using different resolution levels or color/shade combinations.

Screen Data Manipulation

The CRTC generates a special symbol to enable the operator to determine and control the location onscreen where data is to be entered. This symbol is called a `cursor`, and may be any symbol that can be used to indicate position onscreen. The most common cursor symbols are underscores, squares, and triangles. These symbols may be placed under, over, or on top of a character position onscreen. The cursor can be moved around the screen under the control of special keys on the keyboard, or some other input device. It can also be made to blink on and off, or turned into an `inverse video display`, to draw attention to some specific screen location. These actions are programmed through its attributes. The current position of the cursor is held in a special cursor register in the CRTC. Figure 9.10 depicts the dot pattern representations of the three most common cursor shapes.

Figure 9.10

Cursor dot patterns.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1

TRIANGLE

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	1	1

UNDERSCORE

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

SQUARE

Inverse video displays are achieved by inverting the dot-pattern logic levels so that dark characters are displayed on a bright screen. This provides a means of highlighting text or graphics onscreen. The CRTC may also provide the following functions:

- Produce dark characters on a gray background, by substituting a gray-level voltage for the white level

- Cause the display to flash on and off, by alternately applying 1 V and 3 V levels to the light areas of the screen
- Cause the display to pulse, by alternately applying 1 V and an intermediate gray-level voltage to the light areas of the screen

All these functions are provided to allow the user to accentuate desired areas of the display.

To accommodate text editing, the screen memory may be large enough to hold several pages of text. A page of text is expressed as the capacity of the screen. This allows the display to be scrolled up or down. When the screen is full of text, the CRTC will move the top line of text upward (off the screen) to create a vacant line at the bottom of the screen for new data, or to display lines of data that were written in the memory below the current bottom line address.

The key to `scrolling` operations is the `top-line register`, which holds the address of the first line of data being displayed onscreen. Figure 9.11 illustrates the concept behind paging and scrolling. When the data onscreen is to be scrolled up, the register is just incremented to change the address of the first line to be displayed to that of the second display line. In this manner, it appears that each display line is shifted up one line. The display can also be scrolled down by decrementing the top-line register.

Scrolling generally involves one line of text at a time. However, an entire page of text can be scrolled at once. This operation is referred to as `paging`. The display may be paged-up or paged-down just by incrementing or decrementing the top-line register by the proper number of addresses. This is a particularly convenient method of changing display data, because the data is never actually shifted in the memory; only the address of the first display line of the screen is changed. Thus the screen memory addressing logic is kept fairly simple.

Figure 9.12 depicts the functional relationship of a commercially available CRT controller, screen memory, and control logic unit.

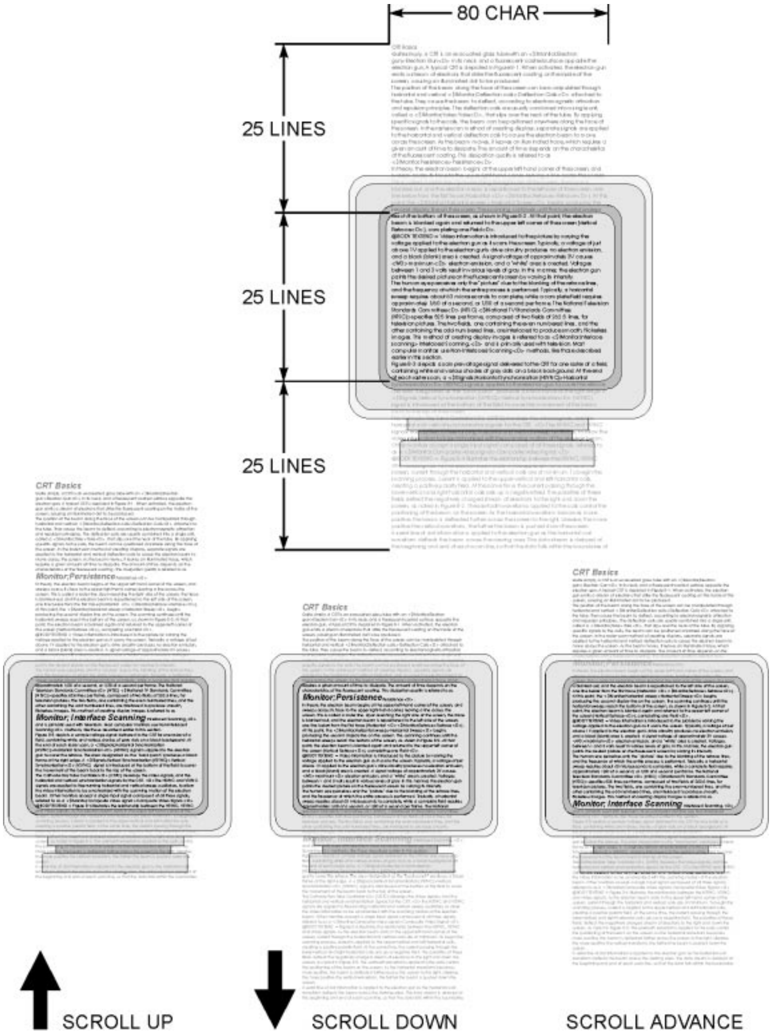


Figure 9.11

Paging and scrolling.

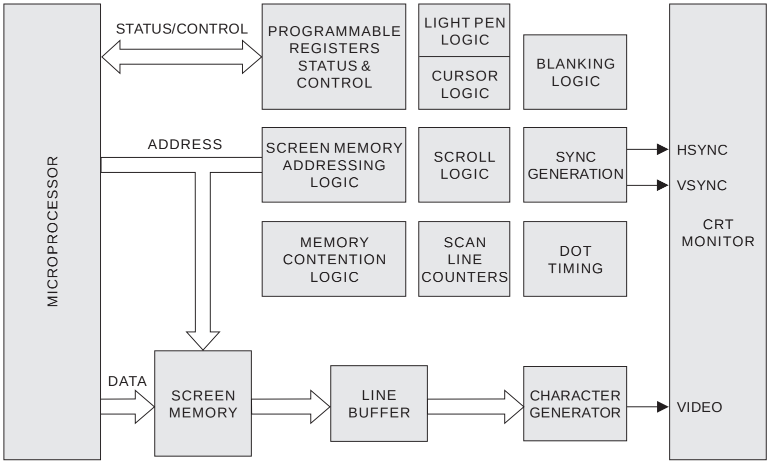
Video Standards

Objective

Many different video display standards have been developed for the IBM PC series and their clones. Each standard uses a different connector or pin arrangement to transfer video display and control information between the video adapter card and the monitor. Of course, the monitor and the adapter card must be compatible with each other.

Figure 9.12

The monitor, CRT controller, video memory, and microprocessor.



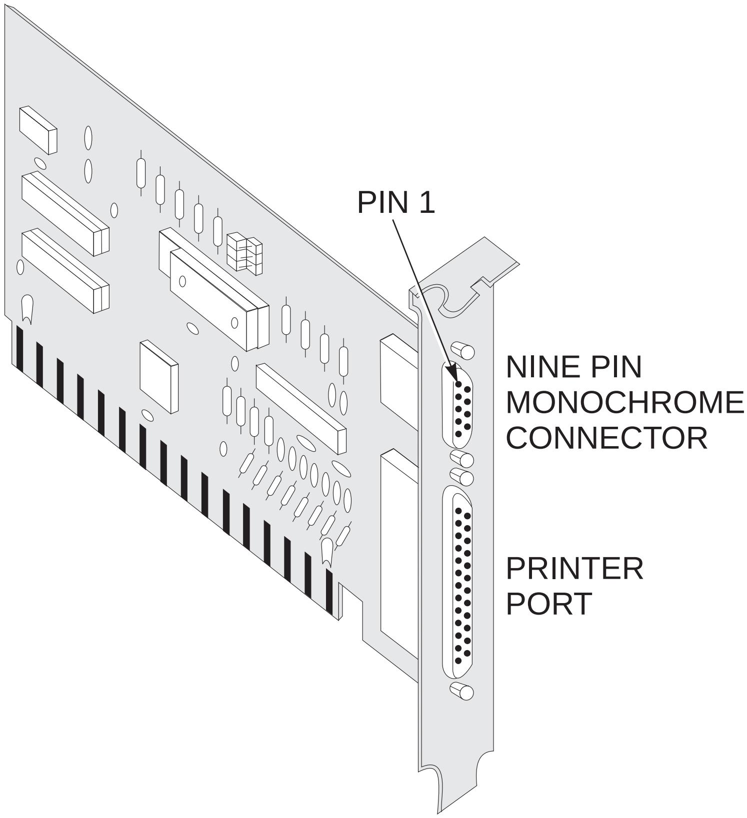
Monochrome/Display/Printer Adapter (MDA)

The `monochrome/display/printer adapter` (MDA) allows PC-compatible monochrome monitors to be attached to the system unit, through a 9-pin, D-type connector. These monitors differ from other composite monitors in that the HSYNC and VSYNC signals are not sent to the monitor along with the video information. A typical MDA adapter is depicted in Figure 9.13.

The MDA card supports an 80-character×25-row screen display in `alphanumeric (A/N) mode`. In this mode, characters are defined as being 7×9 dot patterns contained within 9×14 character cells. The character cell is made larger than the character itself, to create a border between consecutive characters and to allow for characters with `descenders` (portions of the character that hang below the character line). The adapter's ROM character generator is capable of producing 256 different character codes, including the complete standard ASCII character set. It can also produce an extended ASCII character set, which includes special support characters for games, word processing/editing, foreign language, and block-graphics applications. In addition, the adapter supports character attributes of blinking, reverse video, bold face, blanking, and underlining, on a character-by-character basis.

Figure 9.13

The monochrome card.



In the graphics mode, the adapter's operation is `bitmapped`, with each bit of the two pages in the screen memory corresponding to a screen dot. These two pages can be displayed alternately. While one page is being displayed, alterations to the buffer for that page will be displayed immediately. Changes to the page not being displayed will be shown only when that page is selected.

Under these conditions, each page has a resolution of 720 (horizontal) by 348 (vertical) dots (pixels). All the adapter's outputs are TTL-compatible, with the HSYNC signal being a positive-level signal and the VSYNC signal being an inverted, or negative-level signal. The video output signal is semicomposite (the HSYNC and VSYNC signals are separate) and operates in noninterlaced mode with a refresh rate of 50/60 Hz. The MDA adapter has a 9-pin, female D-shell connector for video output, and a 25-pin, D-shell for the LPT1 printer output.

Color Graphic/Printer Adapter (CGA)

The first color display offered for the IBM line of computers was the IBM Color Graphics display. This display was introduced to make the IBM PC competitive with the home entertainment computers of the day that used color television sets for video display purposes. As a matter of fact, the original IBM color graphics adapter (CGA) card included an RF-modulated output port and a composite color output port, which allowed it to be connected to a television set.

The CGA standard sets the monitor's horizontal sweep frequency at 15 KHz and its vertical refresh rate at 60 Hz. Within these confines, the standard produces 7×7 dot characters in an 8×8 character box. The CGA screen format for text accommodates 80 characters across the screen, with 25 character lines down the screen. The standard also allows for a 40-column operation. Although this resolution (640×200) is considerably lower than that found in MDA and MGA cards, the main reason for the introduction of the standard was to provide color. The CGA standard can produce 16 different user-definable colors. The programmer can generate up to sixteen different character colors and eight different background colors. These functions are controlled by software through one of the registers in the adapter's 6845 video controller IC.

In addition to its text mode, the CGA standard provides the capability to produce graphics onscreen. This feature was not available with the original MDA standard. Under that standard, only text characters could be displayed onscreen. However, the CGA standard allows two color graphics modes to be used: low-resolution and medium-resolution. A high-resolution graphics mode is possible, but only in monochrome.

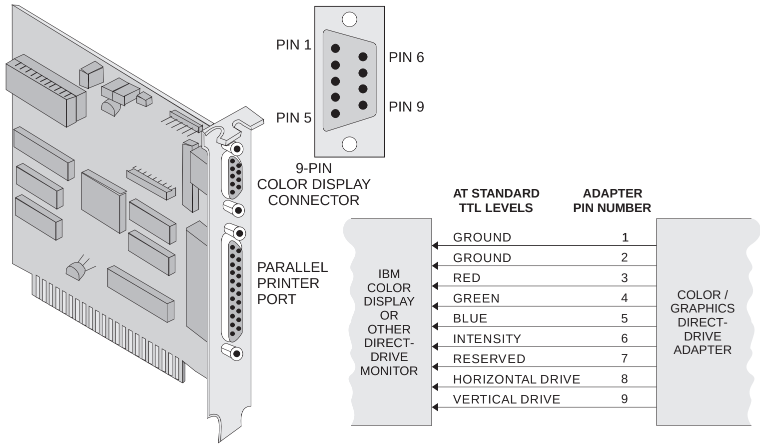
In low-resolution mode, the screen is divided into 100 rows of 160 PELs. Each PEL is two dots high by two dots wide, and can be assigned any one of 16 colors. Medium graphics mode supports 200 rows of 320 PELs each. In this mode, the PELs are one dot high by one dot wide, and can be assigned one of four colors. High-resolution mode produces 640 PELs by 200 rows, with each

PEL mapped directly to a bit of memory. Because each bit is tied to a bit in memory, only monochrome operation is possible in this mode.

A CGA card, displayed in Figure 9.14, provides two output ports at the rear of the unit. These are the RGB color video output, and the parallel printer output (the RF-modulated and composite color outputs of the original CGA card have been discontinued in most later versions of this card). The smaller, 9-pin, female D-shell connector is used to transfer video signals to a color monitor; and the 25-pin, female D-shell connector provides a parallel printer interface connection. This card also includes a light-pen interface from a 4-pin, BERG strip located on the side of the board. Like the MDA card, the CGA's printer port will act as the primary printer port (LPT1) in the system, if not disabled.

Figure 9.14

The CGA card.

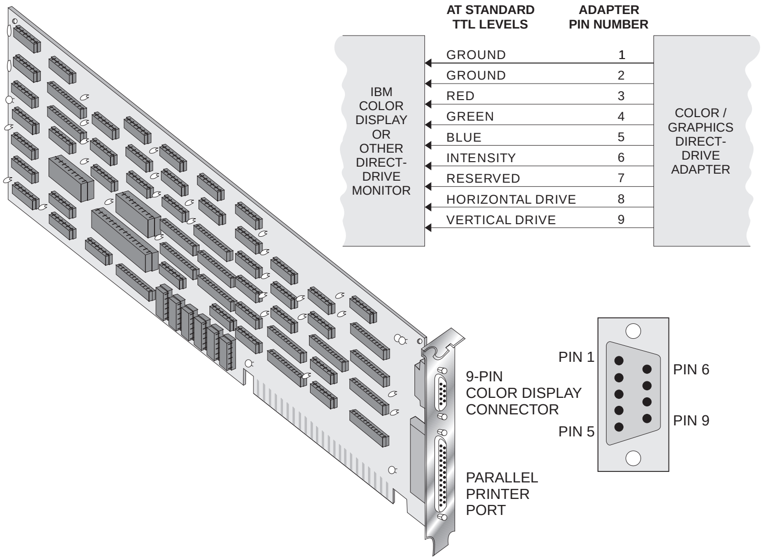


Hercules Graphic/Printer Adapter (HGA)

The Hercules Monochrome Graphics Adapter (HGA) card, illustrated in Figure 9.15, was developed to incorporate the best aspects of the IBM MDA and CGA cards. It combines the bitmapped graphics capabilities of the CGA card with the high-resolution character-mapped text capabilities of the MDA card. It not only matches the MDA's 80×25 character format, but it increases the resolution in graphics mode to 720×348. To accommodate these levels of resolution, the HGA's horizontal sync frequency is increased to 18.1 KHz, with a 50 Hz vertical retrace rate.

Figure 9.15

The HGA card.



The adapter supports three modes of operation: a high-resolution alphanumeric (A/N) text mode, and two all points addressable (APA) graphics modes. When the unit is first powered up, the adapter is in the text mode and its graphics capabilities are masked, so no graphics software can be run. In the graphics modes, the adapter supports 64 KB of video information in an onboard video memory. This 64 KB buffer is divided into two 32 KB buffers for each of the two graphics pages.

In the first of the graphics modes, called the HALF configuration, the first graphics page located at addresses between B0000h and B7FFFh are accessible to graphics software. The second graphics page, located at addresses between B8000h and BFFFFh, is suppressed. This allows other video cards, such as a color/graphics adapter, to be used in the PC as long as their screen buffers do not occupy any part of the first graphics page addresses.

In the second graphics mode, called the FULL configuration, both pages of the adapter's screen buffer are available to graphics software, and other video adapters may not be used in the system.

Enhanced Graphics Adapter (EGA)

The resolution of the CGA card proved to be too low for the tastes of many users. By 1984, IBM had developed a replacement video standard called the `Enhanced Graphics Adapter` (EGA) standard. The EGA standard defines text characters as 7×9 matrix in an 8×14 dot box (640×350 dots). The resolution factor remained constant when the display was shifted into graphics mode. The EGA adapter/monitor interface connection incorporated new signals that allowed up to 64 colors to be used.

To accommodate the high resolution and expanded color possibilities, the EGA card, depicted in Figure 9.16, incorporated a bank-switching video memory scheme. Its memory was divided into four banks (16 KB or 64 KB each) that could be switched (redirected) into the system's video address range. To achieve compatibility with software written for the MDA and CGA standards, the EGA's memory could be switched to the base video memory address of these standards (B0000h for MDA and B8000h for CGA). The base address for EGA video memory begins at hex address A0000h.

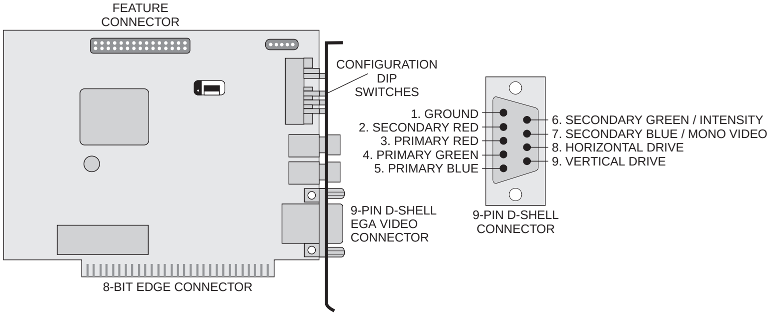


Figure 9.16

An EGA card.

The high-resolution requirements of the EGA standard also called for an increase in the horizontal scan rate. The horizontal scanning frequency in the EGA standard was elevated to 22.1 KHz, with a vertical refresh rate of 60 Hz. However, the EGA card also had to be capable of producing horizontal and vertical refresh rates that were compatible with previous standards.

Unlike its predecessors, the general oversight of the EGA card's operation was not dependent on the system software. Instead, the EGA standard made use of an onboard `EGA video BIOS` routine to manage the board's configuration, mode-control, and EGA compatibility functions.

The EGA card was compatible with a number of different video display monitors. In general, EGA cards supported the MGA standard (and could therefore be used to drive monochrome monitors) as well as the CGA standard (allowing it to be used with RGB color monitors). To avoid damaging the monitor and the adapter, the EGA card had to be configured to work with the type of monitor being used with the system. This was done by setting DIP switches to reflect the type of monitor in use.

The EGA video signal was passed to the monitor through the 9-pin, female D-shell connector on the back plate of the card. This allowed compatibility with the previous MGA and CGA connections. On closer examination, it could be seen that some additional signals had been added to the interface. In particular, secondary intensity signals were added to each of the three color signals. The two color lines worked together in a digitally coded manner to furnish four levels of brightness for each color.

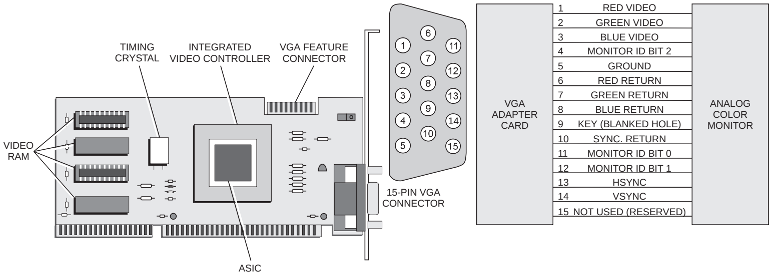
Video Graphics Array (VGA) Adapters

☑ Objective

The next video standard improvement appeared when IBM announced its new line of personal computers that it named `Personal System 2` (PS/2). This video standard acted as one of the centerpieces of this system and was called the `Video Graphics Array` (VGA) standard. In the PS/2 line this function was built directly into the system board in the form of a `very large scale integration` (VLSI) logical gate-array IC. This sometimes led to the VGA acronym being incorrectly expressed as "video gate array." In a short time, many PCB manufacturers were producing options adapter card versions of the VGA controller. Like other options adapter cards, these units were designed to fit into the expansion slot of a typical PC-compatible computer. It allowed VGA-compatible monitors to be connected to the system through a 15-pin, D-shell connector. Figure 9.17 shows a VGA card.

Figure 9.17

A VGA adapter.



VGA Specifications

In designing the VGA standard, IBM departed from the signal formats found in its previous display standards. To accommodate a wide range of onscreen color possibilities, the VGA standard resorted to the use of `analog video signals`. Therefore the intensity of the image could be varied infinitely over the entire voltage range of the signal (unlike digital signals that can only be on or off). This allowed the VGA standard to support up to 262,144 (256 KB) colors.

The digital signals used by the rest of the computer system were converted into analog signals on the video adapter card by circuits called `digital-to-analog converters` (DACs). Nearly all VGA cards employed a three-DAC circuit in their VLSI chip to perform this task (one DAC for each of the three video color signals). In addition, the DAC chip contained the adapter's 256 KB color palette and circuitry, for assigning up to 256 of those colors to be available for use onscreen at any given time.

In addition to offering vastly improved color production capabilities, VGA provided superior resolution capabilities. Standard VGA resolution is defined as 720×400 pixels using 16 colors in text mode, and 640×480 pixels using 16 onscreen colors in graphics mode. However, improved-resolution VGA systems, referred to as `Super VGA's`, are now commonly available in formats of 1,024×768 with 256 colors, 1,024×768 with 16 colors, and 800×600 with 256 colors. The SVGA definition continues to expand, with video controller capabilities ranging up to 1,280×1,024 (with reduced color capabilities) currently available in the market.

IBM produced its own extended graphics array standard, called the XGA. This standard was capable of both 800×600 and 1024×768 resolutions, but added a 132-column, 400-scan line resolution. Unfortunately, IBM based the original XGA on interlaced monitors, and therefore never received a large following.

The maximum resolution/color capabilities of a particular VGA adapter were ultimately dependent on the amount of onboard memory the adapter had installed. The standard 640×480 display format, using 16 colors, requires nearly 256 KB of video memory to operate (640×480×4/8=153600 bytes). With 512 KB of video memory installed, the resolution can be improved to 1024×768, but only 16 colors are possible (1024×768×4/8=393216 bytes). To achieve full 1024×768 resolution with 256 colors, the video memory has to be expanded to a full 1 MB 1024×768×8/8=786432 bytes). Access to this memory is very flexible.

The VGA card's memory-mapping capabilities allow the memory to be accessed from hex base addresses B0000 and B8000 to achieve MDA and CGA compatibility, or at base address A0000 for EGA and VGA standards. Like the EGA standard, the full range of the VGA's installed video memory is divided into banks to accommodate bank-switched memory techniques. Under the VGA standard, there are also plug-and-play (PnP) VGA cards that can automatically adapt to the type of monitor they are connected to. This function is made possible by the addition of four ID lines (monitor ID bits 0–3) to the VGA interface port.

Standard VGA monitors employ a 31.5 KHz horizontal scanning rate; Super VGA monitors use frequencies between 35 and 48 KHz for their horizontal sync, depending on the vertical refresh rate of the adapter card. Standard VGA monitors repaint the screen (vertical refresh) at a frequency of 60 or 70 Hz; Super VGA vertical scanning occurs at frequencies of 56, 60, and 72 Hz. Figure 9.18 presents a summary of the different video standards.

VGA Cards

Early non-IBM VGA adapter cards included a special auxiliary video/feature connector to remain compatible with the IBM function.

Because the microchannel architecture was patented, the third-party manufacturers resorted to placing the connector elsewhere on their cards. Some cards used a notched edge connector along the top of the card; others employed a BERG connector for this function. Although the physical connector was different, the function was the same as the PS/2 versions of the card. They allowed add-on units such as a secondary monitor or a video capture card to share the VGA signals.

Figure 9.18

Video standards.

Standard	Year Introduced	Mode	Resolution (hxv pixels)	A/N Display	A/N Character	Refresh Rate	Horizontal Sweep Rate	Buffer Address
MDA (Monochrome Display Adapter)	1981	Alpha Numeric (A/N)	720 x 348 720 x 348	80 x 25	7 x 9 in 9 x 14	50/60	(Non-interlaced	B0000–B7FFF B0000–B7FFF B0000–BFFFF
CGA (Color Graphics Adapter)	1982	(A/N) Low-resolution Medium-resolution High-resolution (APA) All Points Addressable graphics	640 x 200 160 x 100 320 x 200 640 x 200	80 x 25	7 x 7 in 8 x 8	60	15 kHz	B8000–BBFFF
HGA (Hercules Graphics Adapter)	1982	A/N Diag/Half/Full Graphics	720 x 348	80 x 25	7 x 9 in 9 x 14	50	18.1 kHz	B0000–BFFFF
EGA (Extended Graphics Adapter)	1984	A/N Graphics	640 x 350 640 x 350	80 x 25 80 x 43	7 x 9 in 8 x 14	60 Hz	22.1 kHz	0A0000
VGA (Video Graphics Array Adapter)	1989	Text Graphics	720 x 400 640 x 480	80 x 25 80 x 43	9 x 16	60 or 70 Hz	31.5 kHz	0A0000–0BFFFF
Super VGA (SVGA)	1990	Text Graphics	1280 x 1024 1024 x 768 800 x 600	80 x 25 80 x 43	9 x 16	50, 60, or 72	35-48 kHz	0A0000–0BFFFF
XGA	1992	Text Graphics	1024 x 768 800 x 600	132 x 25	9 x 16 8 x 16	44/70	35.5 kHz	0A0000–0BFFFF

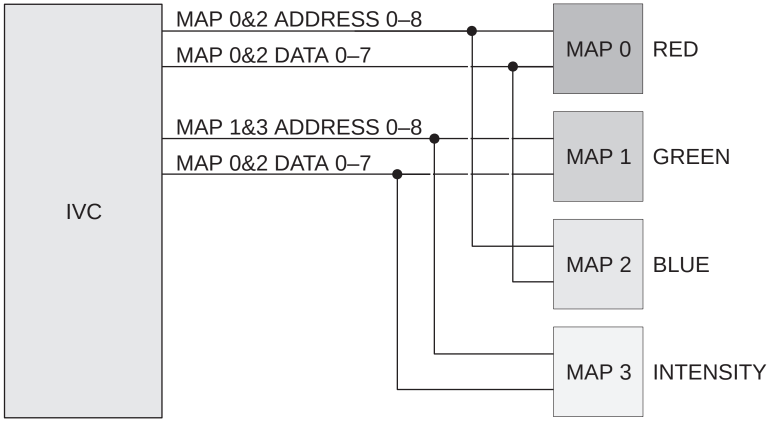
Nearly all VGA adapters were based on an ASIC device called the `integrated video controller (IC)`. A block diagram of the IVC is depicted in Figure 9.19. It was a single-chip, high-performance, low-cost VGA chip. This single IC contained all the circuitry required to generate the video signal going to the monitor connector, and to generate the timing signals for the video memory. It combined a video DAC, an IBM-compatible `video BIOS`, a `CRT controller`, a `graphics controller`, and an `attribute controller` into a single IC device. On older VGA cards, portions of these units were constructed using discrete ICs.

On most newer VGA cards, the only other ICs on the card were the video RAM ICs. They were normally capable of occupying addresses A0000h to BFFFFh; the ROM BIOS held addresses C0000h to C7FFFh. Video cards came with various levels of video

RAM installed. The minimum had typically been 256 KB, but the standard for some time has been 1 MB. Advanced video cards may include 2 MB or more of video RAM.

Figure 9.19

The blocks of the integrated video controller.



The video RAM is normally accomplished through Dynamic RAM devices. However, newer memory types are beginning to offer improved video memory. These memory types include `Extended Data Out DRAM` (EDO RAM), `Synchronous DRAM` (SDRAM), and `Video RAM` (VRAM). EDO RAM memory operates faster than normal DRAM used in screen memories by not disabling its data output between accesses. Therefore the turn-on/turn-off times for the outputs are negated. SDRAM uses an external pixel clock to perform high-speed synchronous data accesses.

VRAM employs a special `dual-port access` system to speed up video operations tremendously. In other memory devices, access to the data in the RAM locations is shared between the system's microprocessor and the adapter's video controller. The microprocessor accesses the RAM to update the data in it, and also to keep it refreshed. Meanwhile, the video controller must access the same locations to move pixel information to the screen. Both control devices must use the same data bus to get to the data.

Unlike the single data port operation found in the other DRAM devices, VRAM devices provide two access buses to the data. The first is a standard set of parallel data bus pins. The second is a special serial port that allows the data to be accessed for refreshing purposes. VRAM tends to be more expensive than simple DRAM chips, but it delivers much faster data access.

Memory speed is important during complex screen operations such as scrolling or word wrapping. In graphics mode, memory speed is particularly important because the entire screen may have to be rewritten bit by bit. If these operations occur too slowly, annoying blinks and flashes are produced onscreen. The access time of the DRAM chips used with the VGA adapter should be no longer than 100 nanoseconds.

Integrated Video Controllers

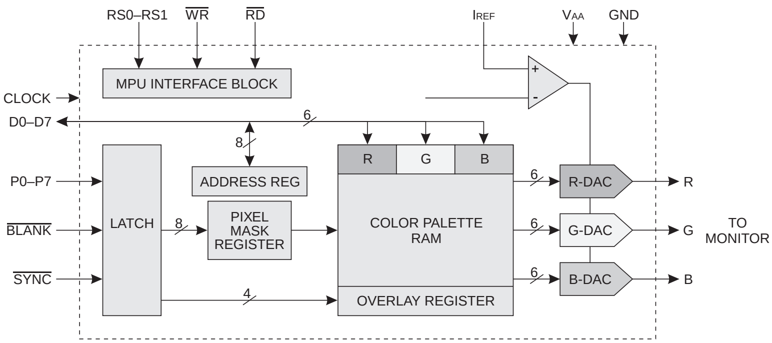
Objective

The integrated video controller receives address and data directly from the expansion bus connector. The system applies addresses to the IVC on the system address (SA0–SA19) lines. Likewise, data moves between the chip and the system on data bus lines (SD0–SD7).

The video controller uses a separate address and data bus for each set of DRAM chips. Most IVCs can address video memory using several different formats. For VGA or EGA operation, video RAM is divided into four sections called `maps`. The four maps represent the three colors—red, green, and blue—and intensity. Figure 9.20 illustrates how the controller addresses these maps.

Figure 9.20

VGA video RAM addressing.



Most VGA adapters come from their manufacturer with a disk full of enhanced software display drivers. These drivers modify the parameters in the registers to optimize the adapter for different software applications such as AutoCAD, Microsoft Windows, and so on. Windows and Windows 95 come with a variety of video drivers already built into them.

RAMDAC Section

The RAMDAC section contains 576 bytes of RAM arranged in a 256×18 format. It also contains three 6-bit DACs. The 6-bit digital values are applied to the DACs' inputs. The DACs convert the coded value into corresponding analog output levels. The RAMDAC contains a RAM look-up table called a `color palette`. This table is used to translate a digital value into a corresponding color using a mixture of red, green, and blue. Remember that all the colors in a color CRT are produced by varying the intensity of the red, green, and blue dots onscreen.

Because each color is a six-bit value, there are 64 levels ($2^6=64$) for each basic color. When the red, green, and blue signals are combined, each color produced onscreen relates to an 18-bit value. This produces a maximum of 262,144 ($2^{18}=262,144$) or 256 KB colors.

When the RAMDAC section receives the color-palette information from the video controller, it stores it in the RAM locations. These values are applied to a DAC to produce the analog video signal for each color. The outputs of the DACs connect directly to the monitor via the adapter's VGA connector. Figure 9.21 shows the RAMDAC section of the video controller.

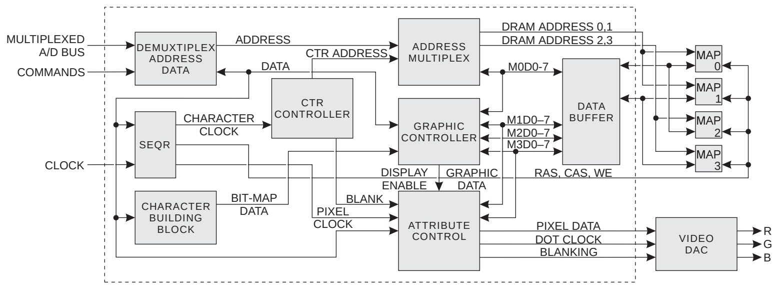


Figure 9.21

RAMDAC block diagram.

Video BIOS Section

The operation of the VGA adapter is controlled by the card's MASK ROM, which provides the video BIOS function calls. This BIOS acts as an extension of the system BIOS, and is located between address C0000h and C7FFFh. All the video function calls are accessed through software interrupt INT10. The video BIOS supports 27 distinct modes of operation. These modes include various character-box sizes and resolution selections. It also includes two different methods of storing screen data in the video memory. The first method is called alphanumeric (A/N) mode, which is used for text operations; the second is referred to as all points addressable (APA) mode, which is normally used for graphics applications.

The BIOS contains the default font styles for the display's character sets. In alphanumeric modes, characters are passed to the screen memory in ASCII-coded format. In graphics modes, however, the adapter's operation is bitmapped, with information concerning each pixel of the screen being stored in the memory. There are three possible starting addresses for the screen memory—$A0000_h$, $B0000_h$, and $B8000_h$. These addresses correspond to the pixel at the top-left corner of the screen for a given mode of operation. When simulating MDA operations, the screen begins at $B0000_h$. CGA simulation begins at $B8000_h$, and EGA/VGA modes begin at $A0000_h$.

Video Controller Section

Consider the block diagram of the video controller section. This section consists of the sequencer (SEQR), a cathode ray tube controller (CRTC), a graphic controller, and an attribute controller.

The Sequencer

The `sequencer` generates basic DRAM timing and the character clock for video memory timing control. The VGA adapter's screen memory is known as `dual-ported RAM` memory, because both the system microprocessor and the video controller can access it. However, both devices cannot access the memory at the same time. A small problem arises when the system accesses the screen memory for an update. The system access requires more time to complete than the interval between sending character codes to the screen.

The sequencer uses latches and timing circuitry to help the system microprocessor access the screen memory during the active retrace interval of the electron beams. This is accomplished by inserting the system's read and write cycles of the screen memory between the video controller access cycles.

The CRTC

The `CRTC` generates the vertical and horizontal synchronization signals (VSYNC and HSYNC) for controlling the electron-gun movement. It also inserts the cursor and underlining for the text modes. It generates the address for the graphic controller to retrieve pixel data from the DRAM display memory for both the text and graphics modes. The video RAM refresh controller function is also performed by this section of the chip.

The Graphics Controller

The `graphics controller` section performs two functions. First, it provides the interface between the DRAM memory and the system's microprocessor when the display memory is accessed by the microprocessor. Second, in graphic modes it formats the pixel data for the attribute controller during the active retrace intervals. When the display is used in text mode, the data bypasses the graphics controller and is applied directly to the attribute controller.

The Attribute Controller

The last major section of the video controller is the `attribute controller`. This section receives data from the display memory (with the aid of the graphics controller) and formats it for the display. It also implements the cursor, underlining, and blinking attributes in the text modes.

Initialization

It should be apparent that the VGA adapter requires a large number of registers to operate. Most of these registers are used to produce a certain mode of operation. Fortunately, the majority of this information is supplied to the adapter through a software interrupt—INT10. During the system startup, the video BIOS intercepts the system's software INT10 function and takes it over by substituting its own enhanced `INT10 function`. When this has been accomplished, the BIOS initializes the VGA adapter's other functions. Three of the more notable setup functions performed by the BIOS are the establishment of the video memory's video maps, the production of the adapter's RAM character generators, and the initialization of the RAMDAC's color palette.

The VGA adapter uses two RAM character generators. By using RAM character generators rather than the ROM generator described earlier, user-defined characters can be employed. When the system is started, default character fonts are downloaded from the video BIOS into the RAM character generators. Both character generators are located in memory map 2, and each consists of seven possible 8 KB character sets. These character sets can be easily rewritten by software at any time.

The system loads the default color values into the RAMDAC's color palette at startup. Three 6-bit color values are written into each palette location—one each for the red, green, and blue value. A write operation into a specific palette position begins by writing the address of the palette position, and is followed by three sequential writes into the DAC data register—red, green, and blue. These write operations must be performed in order, and they must be completed. If the sequence is interrupted before it is completed, the entire operation must be performed again. An auto-increment function causes the value of the write address register to be increased by one, after a complete write sequence has been performed. This allows consecutive updates of the color palette locations, which allows the system to supply continuous red, green, and blue data until the entire palette has been written.

Text Mode

Objective

In text mode, the adapter accepts screen character information from the system's data bus and stores it in the video screen memory. In this mode, the system sends two bytes of information to the adapter for each character to be displayed onscreen. These bytes are stored in sequential order, one after the other. The first byte is the ASCII-coded character. This information is stored as even-address information in memory map 0.

The second byte, called an `attribute byte`, contains information about how the character is to be displayed. This byte is stored as odd-address information in memory map 1. Bits 0–2 of the attribute byte determine the `foreground` (text) `color`. Bits 4–6 determine the `background` (character box) `color`. Bit 3 indicates a highlighted (bold) character, and bit 7 indicates a blinking character. Bit 3 can also be defined to switch between the character sets stored in map 2. In like manner, bit 7 can be redefined to carry background intensity information.

All the VGA adapter's text modes support 16 colors—eight background colors and eight foreground colors, as determined by the character's attribute byte. The serialized character dots are multiplexed with the attribute byte. If the value of the dot is a logic 1, indicating a foreground color, the foreground color bits of the attribute byte (0–3) are read and applied to the color plane enable logic. If the character dot is a logic 0, the background color bits (4–7) are used to drive this circuitry. In the case of 9-dot character cells, the ninth bit is automatically set to the same value as the background color. Table 9.1 lists the default color values loaded into the palette registers by the video BIOS.

Table 9.1

The video BIOS color set.

Intensity	Red	Green	Blue	Color
0	0	0	0	Black
0	0	0	1	Blue

continues

Table 9.1 Continued

Intensity	Red	Green	Blue	Color
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Gray
1	0	0	1	Light blue
1	0	1	0	Light green
1	0	1	1	Light cyan
1	1	0	0	Light red
1	1	0	1	Light magenta
1	1	1	0	Yellow
1	1	1	1	White (high intensity)

The 8-bit pixel address information from the attribute controller (P0–P7) is latched into the `pixel mask register` of the RAMDAC section. The output of the mask register is used to access one of the 256 color addresses in the color palette. The selected color palette address causes the three, 6-bit pieces of color information in that location to be applied to the three DACs.

A value of $3F_h$ will drive the DAC to a current output value of 17.62 mA. This corresponds to a level of white. Conversely, a value of 00_h produces a 0.0 mA output current that corresponds to a `black level`. Other hex values applied to the DAC input produce an output current that is equal to the numerical value of the input plus 9.05 mA. The presence of an active BLANK signal overrides the code value applied to the DAC, and produces a 0.0 mA black level.

Along with the pixel information, the adapter must furnish accurate horizontal and vertical synchronization signals to the monitor. This function is controlled by the CRTC section. Both signals originate with the CRTC and operate at TTL logic levels. A typical range of the HSYNC and VSYNC signals possible with a VGA adapter was described earlier. Figure 9.22 depicts the VGA adapter's text mode operation.

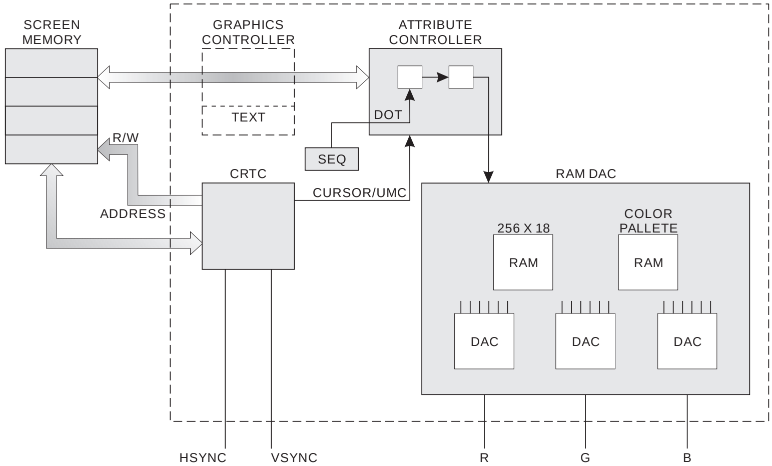


Figure 9.22

Text mode circuitry.

In alphanumeric mode, the adapter supports both 40- and 80-column operations. In 40-column mode, it supports 25 rows of 40 characters. Storing one character byte and one attribute byte per character requires 2 KB of video memory per page. Likewise, a page in 80-column mode will require 4 KB of memory.

If the starting address for a particular 80-column text mode is $B0000_h$ and 2 bytes of memory are required for each character, an 80-character by 25-line display will use memory addresses up to B0F9Fh for the first screen of memory. The space between B0F9Fh and B0FFFh is reserved so that the next page begins at B1000h. If this text mode is designated as having a 9×14 character cell, each scan row will be composed of 720 pixels. The total number of scan rows will be 350. Therefore the resolution of the screen will be 720×350 in this mode.

Graphics Mode

In graphics modes, each pixel onscreen is related to a memory location in each of the adapter's maps. Therefore the adapter does not use the character generator scheme described in text mode operation. Instead, the CRTC loads the screen memory locations' contents into the graphics controller section of the IC. Each pixel's color is determined by combining a bit from each of the adapter's four maps in (16-color mode), or eight maps (in 256-color mode).

The system writes the screen data into the screen memory through the graphics controller, starting at location $A0000_h$ in default VGA mode. Each of the maps correspond to the same sets of addresses.

In 320×200 graphics modes, the display memory is organized into two banks of 8 KB each. Each byte is used to encode four PELS having four possible colors. The beginning address is $B8000_h$, and image maps 0 and 1 are used to hold the image data. This requires 16 KB of video memory. The display is double-scanned to display as 400 rows.

The 640×200 graphics mode uses a single bit plane (C0) that begins at B8000. It uses the same addressing and scan-line techniques as the 320×200 mode. Each bit of the data byte is used to distinguish between two colors for a corresponding PEL.

In the 640×350 mode, the operation of the adapter is expanded greatly. Blank, video, blinking video, and intensified video attributes are added to the operation, increasing the total memory requirement to 56 KB. This mode uses maps 0 and 2, both of which begin at $A0000_h$. The first eight PELs are defined by the byte at A0000, and the next eight PELs by the byte at A0001, and so forth.

The 640×480 mode provides two-color operations using the same memory format described for the 640×200 mode. Instead of starting at B8000, this mode begins at A0000, and no double-scanning is used.

In 16-color modes, the adapter uses all four bit planes, each of which begins at $A0000_h$. Bit plane C0 holds the `blue data`, C1 holds the `green data`, C2 holds the `red data`, and C3 holds the `intensified data`. These values are combined to access the palette register in the attribute controller. In 256-color mode, the video memory starts at $A0000_h$ and uses four maps, each of which is 64 KB long. Each byte contains the color information for one pixel. Each map is sampled twice to produce 8-bit plane values to address the video DAC.

The color data read from the maps is latched and multiplexed in the graphics controller section. The output from the multiplexer is applied to the graphics controller's four `graphics serializers` (C0–C3). Like the alpha serializer, these shift registers accept the data bytes from the screen memory and convert them into serial bit streams. This data is supplied to the attribute controller's alpha serializer. The attribute controller's foreground and background inputs are deselected, so the serial PEL information is clocked directly through the serializer. The operation of the attribute controller and RAMDAC sections are identical to that described for alphanumeric mode. Figure 9.23 describes the operation of the VGA circuitry in graphics modes.

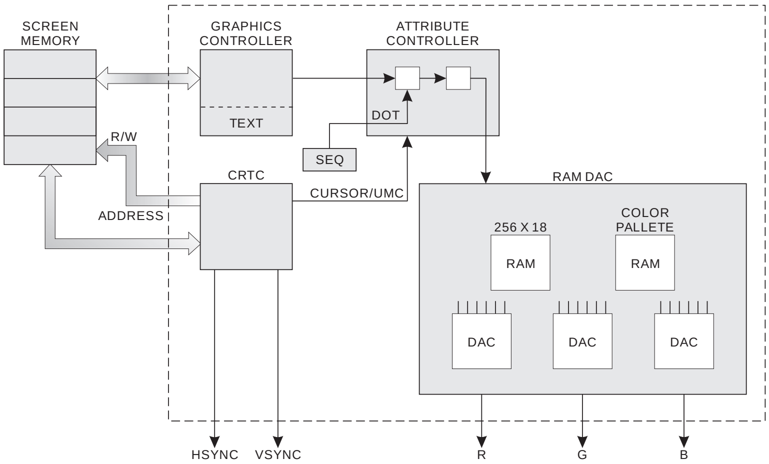


Figure 9.23

Graphics mode circuitry.

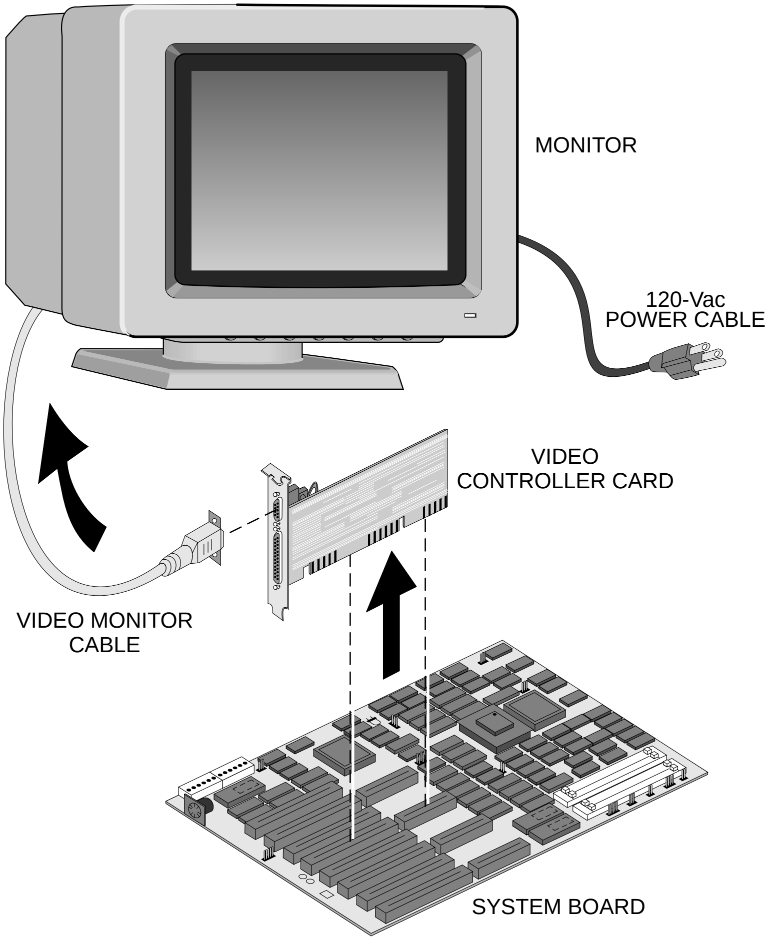
Video Troubleshooting

Objective

The movement of video information is depicted in Figure 9.24. It may be most practical to think of the video information as starting out on the system board. In reality, the keyboard, one of the disk drives, or some other input/output device may be the actual originating point for the information. In any case, information intended for the video-display monitor moves from the system board to the video adapter card via the system board's expansion slots. The adapter card also obtains power for its operation from these expansion slots. Finally, the information is applied to the monitor through the video signal cable.

Figure 9.24

Video information movement.

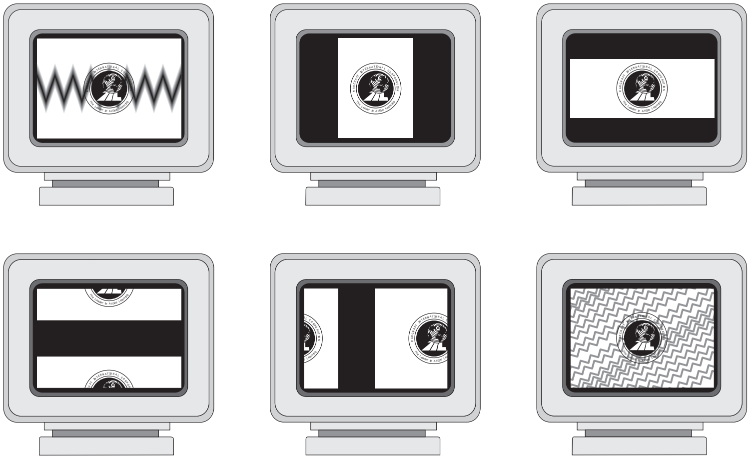


There are basically three levels of troubleshooting that apply to video problems. These are the `DOS level`, the `Windows level`, and the `hardware level`. At the DOS level, there are two considerations: configuration problems and hardware problems. In the case of hardware problems, the components associated with video problems include the video adapter card and the monitor. To a lesser degree, the system board and optional adapter cards, such as sound and scanner cards, can be the cause of video problems.

Figure 9.25 illustrates some of the typical symptoms produced by video display failures.

Figure 9.25

Video failures.



Other common symptoms associated with display problems include the following:

1. No display.
2. Wrong characters displayed onscreen.
3. Diagonal lines onscreen (no horizontal control).
4. Display scrolls (no vertical control).
5. An error code of one long and six short beeps is produced by the system.
6. A `Display Switch Setting Not Proper--Failure to verify display type` error displays onscreen.
7. A `CMOS Display Mismatch--Failure to verify display type` error displays onscreen.
8. An error code of one long and two short beeps indicates a display adapter problem.

In the following troubleshooting sections, the digital portion of the video system is covered. In a later section, the hardware portion of the monitor is taken up. Only experienced technicians should participate in troubleshooting interior monitor problems because of the very high voltages present there.

Configuration Checks

While booting up the system to the DOS prompt, observe the BIOS video-type information displayed on the monitor. The values stored in this CMOS memory must accurately reflect the type of monitor installed in the system; otherwise, an error will occur. These values can be accessed for change by pressing the Ctlr and Del keys (or some other key combination) simultaneously during the bootup procedure.

DOS Checks

Reboot the system, and run a diagnostic software program if possible. Try to use a diagnostic program that conducts a bank of tests on the video components. Run the program's `Video Tests` function and perform the equivalent of the ALL tests function.

Note all the errors indicated by the tests. If a single type of error is indicated, it may be possible to take some corrective actions. If more complex system board problems are indicated, however, exit the diagnostic program, and use the troubleshooting information in the "Hardware Checks" section of this chapter to locate and repair the video problem.

Windows 3.x Checks

If the display operates correctly in DOS, but develops problems when Windows is running, it will be necessary to troubleshoot the Windows structure as it applies to the system's video. Observe the symptoms created. Is an error message produced when Windows is started? If yes, troubleshoot the particular error mentioned in the code. Refer to the `Windows User's Guide` for error-code solution information.

Start the Windows video checks by examining the Windows video driver settings. Incorrect video drive settings can cause symptoms that include blank screens, black borders around the screen, and unreadable displays. At the DOS prompt, type **WIN** to start the Windows program. Double-click the Main icon, and then double-click the Setup icon. Note the video driver listed in the Setup window.

If the video driver from the list is not correct, reload the correct driver. To change the driver setting, click the Options entry from the toolbar, and then click the Change System Settings option from the menu. Use the scroll arrow at the right of the window to move through the available driver options. If the problem persists, reinstall Windows.

If the Windows video problem prevents you from being able to see the driver list, move to the DOS prompt and change directories so that the current directory is Windows. Type and run **SETUP** from the DOS prompt. Install the generic VGA video driver from the list. If the problem goes away, contact the card maker for a new, compatible video driver. If the problem remains, reinstall Windows. If the video is distorted or rolling, try an alternative video driver from the list. If a blank screen appears after Windows starts up, boot the system from a clean Windows boot disk.

To perform deeper Windows video checks, return to the DOS prompt and move to the Windows directory. This is accomplished by typing **CD\WINDOWS** at the DOS prompt. Then type **Edit SYSTEM.INI**, and press the Enter key to examine the SYSTEM.INI file.

Locate the `[Boot]` section heading, near the top of the file, and check the section's video-related parameters. In particular, note the Grabber (.XGR) and Display (.DRV) settings. Also, locate the `[386enh]` section heading, near the top of the file, and check the section's Display parameters. Refer to the `Windows Resource Kit` from Microsoft for information about these settings.

Win95 Checks

Access to the Win95 video information can be found by double-clicking the Control Panel's Display icon. From the Display page there are a series of file folder tabs at the top of the screen. Of particular interest is the Settings tab. Under this tab, the Change Display Type button provides access to both the adapter-type and monitor-type settings.

In the Adapter Type window, information about the adapter's Manufacturer, Version #, and Current driver files is given. Pressing the Change button beside this window will bring a listing of available drivers to select from. You can also use the Have Disk button with an OEM disk to install video drivers not included in the list. The manner in which the list is displayed can also be altered by choosing the Show Compatible Devices or the Show All Devices options.

In the Monitor Type window, there is an option list for both Manufacturers and Models. This function can also be used with the Have Disk button to establish OEM settings for the monitor.

Additional Win95 video information can be accessed under the Control Panel's System icon. Inside the System Properties page, click on the Device Manager tab and select the Display Adapters option from the list. Double-click on the monitor icon that appears as a branch.

The adapter's properties page will pop up onscreen. From this page, the Driver tab reveals the driver file in use. Selecting the Resources tab displays the video adapter's register address ranges and the video memory address range. These settings can be manipulated manually by clicking on the Change Setting button. Information about the monitor can also be obtained through the System icon.

As with the 3.x version of Windows, the first step for isolating Win95 video problems involves checking the video drivers. Check for the drivers in the locations specified in the previous paragraphs. If the video driver from the list is not correct, reload the correct driver. If the problem persists, reinstall Windows 95.

If the Windows video problem prevents you from being able to see the driver, restart the system, press the F8 function key when the "Starting Windows 95" message appears, and select Safe Mode. This should load Windows with the standard 640 x 480 x 16-colors VGA driver. This is the most fundamental driver available for VGA monitors, and should furnish a starting point for installing the correct driver for the monitor being used.

If the problem reappears when a higher resolution driver is selected, refer to the Color Palette box under the Control Panel's Display Option/Settings tab, and try minimum color settings. If the problem remains, reinstall the driver from the Windows 95 distribution disk or CD. If the video is distorted or rolling, try an alternative video driver from the list.

Hardware Checks

If a video-display hardware problem is suspected, the first task is to check the monitor's ON/OFF switch to see that it is in the ON position. Also, check the monitor's power cord to see that it is either plugged into the power supply's monitor outlet, or into an active 120-Vac commercial outlet. Also check the monitor's `Intensity` and `Contrast` controls to make certain that they are not turned down.

If the monitor is using a 9-pin, D-shell connector, the type of monitor and video adapter being used should be checked. Because the MGA, CGA, HGA, and EGA monitors all use this connector, it is possible that they could be interchanged with the incorrect monitor or adapter. This situation can result in damage to both the monitor and the adapter.

The next step is to determine which of the video-related components is involved. On most monitors, this can be done by just removing the video signal cable from the adapter card. If a raster appears onscreen with the signal cable removed, the problem is probably a system problem, and the monitor is good. If the monitor is an EPA-Certified Energy Star Compliant monitor, this test may not work. Monitors that possess this power-saving feature, revert to a low-power mode when they do not receive a signal change for a period of time.

Note

> The process for isolating video problems is reinforced and expanded in the accompanying *Hands-On Lab Book* in procedure 22.

With the system off, remove all the externally connected devices from the system, except for the monitor and the keyboard. Remove any unnecessary options cards from the expansion slots. In particular, remove multimedia-related cards such as sound and video capture cards. Try to reboot the system.

If the system boots up, and the display is correct with all the options removed, it is safe to assume that one of them is the cause of the problem. To verify which external device is causing the problem, reconnect the devices one at a time until the problem reappears. The last device reinstalled before the problem reappeared is defective.

Replace this item and continue reinstalling options, one at a time, until all the options have been reinstalled. If another failure occurs while reinstalling the options, replace that option as well. When all the options have been reinstalled, return the system to full service and service or replace the defective options as indicated.

If the display is still wrong or missing, check the components associated with the video-display monitor. Start by disconnecting the monitor's signal cable from the video-controller card at the rear of the system unit, and its power cable from the power-supply connector or the 120-Vac outlet. Then exchange the monitor for a known good one of the same type (VGA-for-VGA, for example). If the system boots up and the video display is correct, return the system to full service and service the defective monitor as indicated.

If the display is still not correct, exchange the video controller card with a known good one of the same type. Remove the system unit's outer cover. Disconnect the monitor's signal cable from the video controller card. Swap the video controller card with a known good one of the same type. Reconnect the monitor's signal cable to the new video controller card and reboot the system.

Other symptoms that point to the video adapter card include a shaky video display and a high-pitched squeal from the monitor or system unit.

If the system boots up and the video display is correct, replace the system unit's outer cover, return the system to full service, and service the defective video controller appropriately. If the system still does not perform properly, the source of the problem may be in the system board.

The Monitor

Objective

All the circuitry discussed so far is part of the computer or its video adapter unit. The circuitry inside the monitor is responsible for accepting, amplifying, and routing the video and synchronizing information to the CRT's electron gun(s) and the deflection coils.

The RGB monitor accepts and processes separate red, green, and blue signals, which improves the quality of the display by eliminating errors resulting from signal crossovers. The synchronizing signals may be separate, as shown, or they may be carried with one of the color signals (usually green).

Great caution must be used when opening or working inside the monitor. The voltage levels present during operation are lethal. Electrical potentials as high as `25,000 volts` are present inside the unit when it is operating. Operation of a monitor with the cover removed poses a shock hazard from the power supply. Therefore servicing should not be attempted by anyone unfamiliar with the safety precautions associated with high-voltage equipment.

The high-voltage levels do not necessarily disappear because the power to the monitor is turned off. Like television sets, monitors have circuitry capable of storing high-voltage potentials long after power has been removed. Always discharge the `anode` of the picture tube to the receiver chassis before handling the CRT tube. Because of the high-voltage levels, antistatic grounding straps should never be worn when working inside the monitor.

An additional hazard associated with handling CRTs is that the tube is fragile. Extra care should be taken to prevent the neck of the tube from striking any surface. The tube should never be lifted by the neck. This is particularly true when removing or replacing a CRT tube in the chassis. If the picture tube's envelope is cracked or ruptured, the inrush of air will cause a high velocity implosion, and the glass will fly in all directions. Therefore protective goggles should always be worn when handling picture tubes.

Color monitors produce a relatively high level of X-radiation. The CRT tube is designed to limit X-radiation at its specified operating voltage. If a replacement CRT tube is being installed, make certain to replace it with one of the same type, and with suffix numbers that are the same. This information can be obtained from the chassis schematic diagram inside the monitor's housing.

Assembly/Disassembly

Access to the insides of most monitors is gained by removing the back half of the cabinet from the body. The monitor should be placed face down on a soft material to prevent scratching or other damage to the face of the tube. Begin the disassembly by removing the screws that secure the back half of the cabinet to the front. The number of screws involved varies from model to model. Be aware that some screws may be hidden under removable caps. With the screws removed, the back half of the cabinet should lift away, exposing the inside of the unit.

After the monitor's housing has been removed, the internal components can be accessed. The primary components are the CRT tube, the deflection yoke assembly, the high-voltage anode, the power supply, the CRT socket board, and the signal processing circuitry (main board).

The power supply in a monitor is normally a power-supply board with no external cabinet. The function of the board is the same as that of a computer power supply. It receives commercial power, and converts it into the various voltage levels required by the different parts of the monitor. Be particularly careful in dealing with the power supply and the high-voltage anode that plugs into the top of the tube. The voltage levels associated with this connection range between 20,000 and 30,000 volts.

Color monitors also require voltage levels of 4,000 to 6,000 volts for acceleration of the electron beam(s). The current changes involved in going from dark to light images onscreen require high-voltage regulation to prevent unwanted changes in picture size, linearity, and defocusing. A typical monitor's power-supply board is depicted in Figure 9.26.

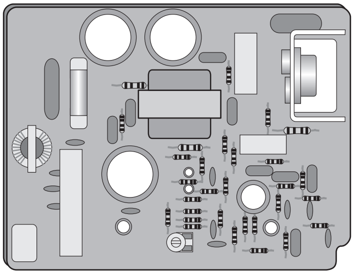


Figure 9.26

A monitor power-supply board.

The `monitor's main board` is responsible for accepting video and sync input signals from the video adapter card and converting them into the signals necessary to control the operation of the picture tube. A main board from a typical color monitor is shown in Figure 9.27. Most of the monitor's service adjustments are located on this board. The monitor's Service/Normal switch is also located here. This switch is used to place the monitor's circuitry in test mode. The actual connection of the high-voltage anode and deflection yoke harness is made on the main board.

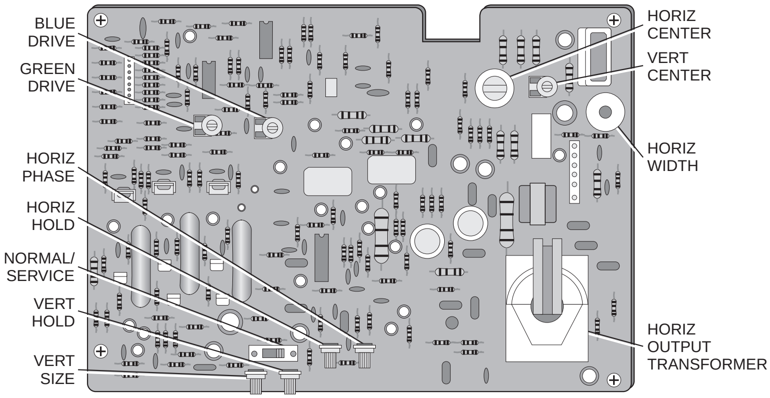


Figure 9.27

A typical monitor main board.

The `CRT socket board`, depicted in Figure 9.28, plugs into the rear of the CRT tube and delivers the video and sync signals to it. It also supplies control voltage to the CRT grids to accelerate the electron beam(s) toward the screen, and to sweep them across its face. On some models, this board contains a portion of the service adjustments.

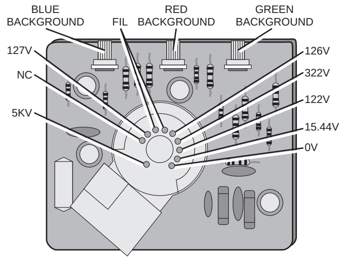


Figure 9.28

A CRT socket board.

The yoke assembly fine-tunes the focus of the electron beams on the face of the screen. In color monitors, the deflection yoke assembly contains several parts. There are two sets of coils in the yoke that orient the picture onscreen, both horizontally and vertically. There are also sets of convergence magnets that fine-tune the alignment of the three electron beams so that they converge into a single dot. In some monitors, the coils and magnets are adjustable, as described in the "Adjustments" section of this chapter.

This coating resides on the inside face of the picture tube and normally provides a return path to ground for electrons shot from the gun(s).

Handling Picture Tubes

If it becomes necessary to replace a picture tube, extreme care should be taken not to damage either tube. To remove the tube, it will be necessary to discharge the high-voltage anode and remove it from the tube. The capacitance of the aquadag coating stores a high-voltage charge that must be discharged before handling the tube and high-voltage anode.

With the power turned off, unplug the monitor's power cable and disconnect the signal cable from the system unit. Next, clip one end of an insulated jumper wire to the chassis ground, and the other end to a screwdriver with a well-insulated handle. While touching only the insulated handle of the screwdriver, slide its blade under the cup of the anode and touch its metal contact. Continue the contact for several seconds to ensure that the voltage has been bled off.

Remove the socket board from the rear of the tube and disconnect the ground leads. Disconnect the deflection yoke circuitry from the monitor's circuit boards. In many monitors, the yoke assemblies are bonded to the neck of the tube, and cannot be removed.

Loosen all mounting hardware that holds the tube to the chassis. Position your hands around the sides of the tube and lift it out of the chassis. Lay the tube face down on a soft, protective surface. As soon as possible, place the tube in a protective housing or a shipping box to reduce the implosion hazard.

Tools

For initial monitor checks, only simple tools are required. However, serious monitor diagnostic and repair work calls for a group of specialized CRT testing tools. Among the specialized tools required are a high-voltage probe, a CRT tester, and an RGB video-pattern generator. The simple tools include a medium Phillips screwdriver, and a 1/4-inch nut driver. Other tools associated with monitor repair include a TV color-bar generator and a degaussing coil.

Adjustments

In addition to the normal external adjustments, monitors typically possess internal coarse adjustments that are not accessible to the general public. These adjustments usually include the Focus and Centering adjustments, and are found on the monitor's main circuit board. To set the horizontal and vertical centering adjustments, it is necessary to connect an RGB video-pattern generator to the monitor. With the generator set to a color-bar pattern, adjust the horizontal and vertical centering settings to obtain the best centering of the raster.

Other common internal adjustments include the vertical hold, vertical size, vertical linearity, horizontal hold, and sub-brightness controls. Additional adjustments are present in color monitors to set the color values and mixtures. Color-related adjustments include red, blue, and green background color settings, as well as red, green, and blue drive adjustments. Over time, the monitor's colors will tend to fade due to the screen's phosphor coating wearing away from use. When this happens, it will become more and more difficult to produce good color output. These controls are also normally located on the monitor's main circuit board.

Figure 9.29 illustrates how incorrect adjustments of the operating and maintenance controls can create trouble symptoms. The misalignment can be caused by an untrained person attempting to service the unit, or by deterioration in the value of a component. These symptoms are correlated to the likely cause in Table 9.2. In each case, the corresponding control should be adjusted to correct the condition. In addition to the symptoms listed in the table, misalignment of the monitor's Automatic Gain Control (AGC) adjustment can cause a weak picture, no picture, or a severely distorted contrast in the picture.

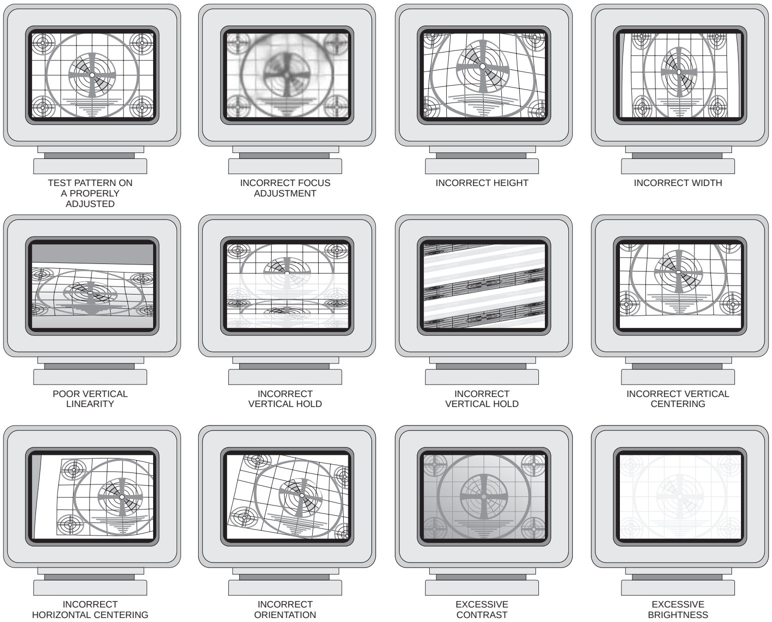


Figure 9.29

Incorrect adjustment symptoms.

Table 9.2

Symptoms/causes.

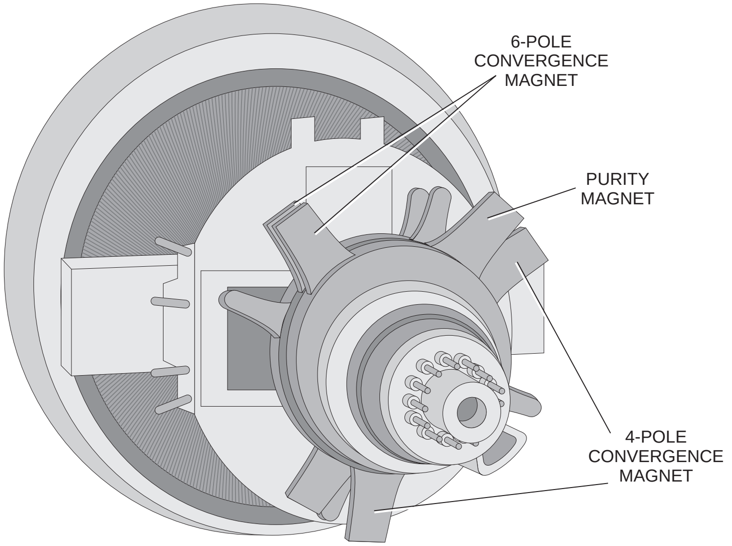
Incorrect Condition	Adjust
Out of focus	Focus
Too tall	Vertical size
Too narrow	Width
Stretched at bottom	Vertical linearity
Picture rolls	Vertical hold

Incorrect Condition	Adjust
Diagonal lines	Horizontal hold
Blank screen above picture	Vertical centering
Blank screen beside picture	Horizontal centering
Slanted picture	Yoke
Picture elements too dark	Contrast
Picture too bright	Brightness

Besides the internal and external operating and maintenance adjustments, the operation of the monitor is affected by a set of magnets in the deflection yoke. A set of `purity magnets` around the neck of the tube provide a color purity adjustment function to produce clean colors. With the blue and green background adjustments turned to minimum and the red background adjustment set so that a red raster is produced onscreen, the deflection yoke is positioned against the purity convergence assembly, as depicted in Figure 9.30. The purity magnets are adjusted until a vertical red stripe is produced at the center of the screen. Then the deflection yoke is moved forward until a uniform red raster is produced. A similar check for both a pure blue, and a pure green, screen should be performed by adjusting the background colors one at a time.

Figure 9.30

The deflection yoke.



A set of convergence magnets in the yoke assembly are used to focus the red, green, and blue dot-trios for proper alignment. Adjusting these magnets requires the use of an RGB video-pattern generator. With the generator attached to the monitor's input jack and tuned to produce a dot pattern, the 4-pole magnets, also illustrated in Figure 9.30, are adjusted so that the red and blue dots converge at the screen's center. After this, the 6-pole magnets are adjusted so that the red and blue dots converge over the green dots.

After converging the dots, tune the generator to a cross-hatch pattern and remove the rubber wedges from between the yoke and the neck of the CRT. Tilt the deflection yoke up or down until the vertical lines at the top and bottom of the screen, and the horizontal lines at the left and right sides of the screen, converge. Then tilt the yoke left or right until the horizontal lines at the top and bottom, and the vertical lines at the right and left sides, of the screen converge. Adjust the yoke for the best convergence possible, and reinstall the rubber wedges.

Some monitors include a `color temperature adjustment` that will produce a white color when all three colors are properly balanced. With the red, blue, and green drive settings adjusted to mid-range, and the red, blue, and green background adjustments set to minimum, set the unit's Service switch to the Service position. Advance the screen control setting from minimum, until a dim line of the predominate color appears. Then adjust the other two background colors until a dim white line is produced. Set the service switch back to normal and adjust the blue and green drive controls for monochrome operation with the best picture quality and high brightness.

Monitor Troubleshooting

Objective

The first step in isolating the monitor as the cause of the problem is to exchange it for a known good one. If the replacement works, the problem must be located in the monitor. Like the keyboard, the monitor is easy to swap because it is external and only involves two cables. If the problem produces a blank display, disconnect the monitor's signal cable from its video adapter card. If a raster appears, a video card problem is indicated.

Obvious items should be checked first. Examine the power cable to see that it is plugged in. Check to see that the monitor's power switch is in the ON position. Check the external settings to see that the brightness and contrast settings are not turned off.

Most monitor defects can be associated with the appearance of the output onscreen. These symptoms can generally be grouped into four general categories. These are power-supply problems, vertical-deflection problems, horizontal-deflection problems, and video problems. If these types of problems occur and cannot be corrected through the monitor's operation and maintenance adjustments, it will be necessary to exchange sections of the monitor's hardware.

Power-Supply Problems

Like the power-supply unit in the computer, the monitor's power supply can affect all the other subunits in the monitor. It supplies B+ voltages for the horizontal and vertical circuits to produce the monitor's raster. The main board derives lower voltage power from the power supply to process the RGB and sync signals. The very high voltages for the HV anode are produced by the power-supply unit.

Check the monitor's internal fuse to ensure that it is good. Before connecting any test equipment to the unit, make sure that the monitor is unplugged and has been discharged.

The blooming effect, depicted in Part-A of Figure 9.31, is an abnormal enlargement of the picture and may involve the monitor's power supply or its main board circuitry.

The stretched appearance of the image in Part-B is a result of a change in the value of the vertical height or linearity value, and can also be attributed to a defective or misadjusted power-supply voltage.

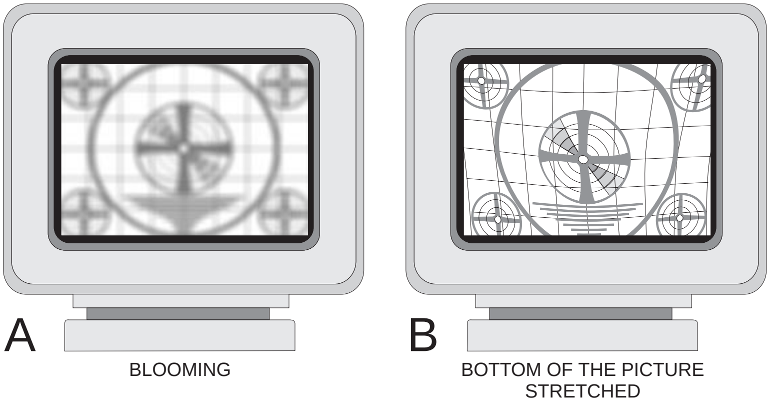


Figure 9.31

Power-supply–related problems.

Vertical-Deflection Problems

Faulty operation of the monitor's sync section will create symptoms similar to those depicted in Figure 9.32. Identical symptoms may be created by the vertical- or horizontal-deflection systems. The horizontal, white-line pattern in Part-A indicates a loss of vertical deflection. The excessive height in Part-B, and the picture that is too short in Part-C, are also vertical height and linearity problems. All these problems relate to the vertical circuitry normally found on the main board. Likewise, the center fold-over, depicted in Part-D and Part-E, is also caused by circuitry on the main board. The double-image condition depicted in Part-F is another vertical-circuitry problem.

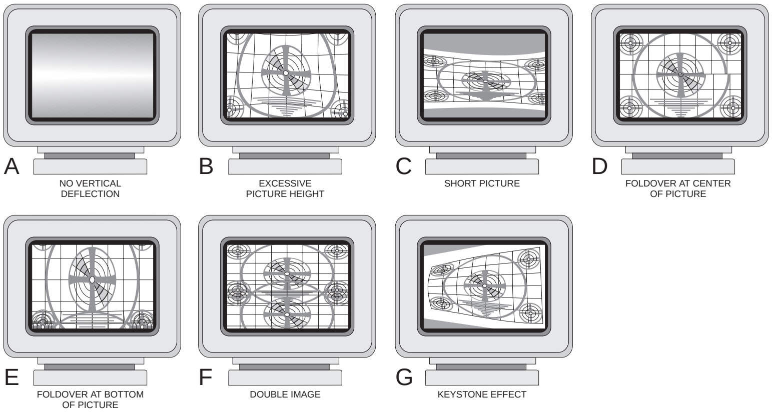


Figure 9.32

Vertical deflection problems.

The monitor's power supply should be checked if this condition exists. The keystone effect, shown in Part-G, is caused by problems with one of the vertical deflection coils in the yoke assembly. The yoke would need to be replaced to correct this problem.

Horizontal-Deflection Problems

When a monitor loses its horizontal sync signal, a symptom such as the one displayed in Part-A of Figure 9.33 is produced. This defect is normally associated with the horizontal sync circuitry located on the main board. Other main-board–related horizontal sync problems include the pie-crust effect shown in Part-B and the right/left half-reversal symptom in Part-C.

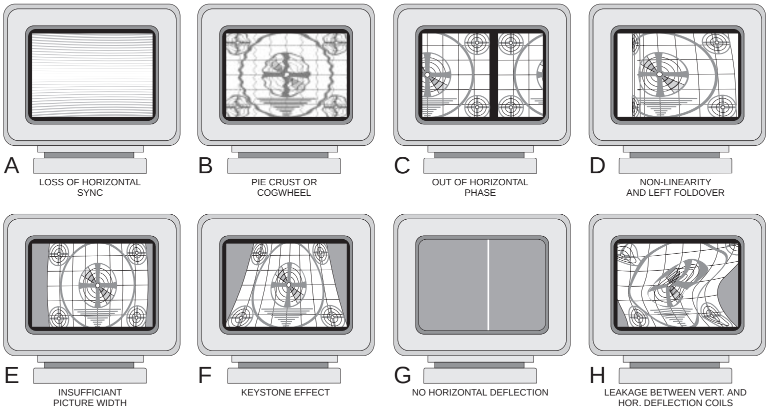


Figure 9.33

Horizontal sync problems.

The horizontal fold-over and insufficient picture-width symptoms, described in Part-D and Part-E, can involve the main board or the deflection yoke assembly. Likewise, the keystone effect caused by a defective horizontal-deflection coil, illustrated in Part-F, will also require that the yoke assembly be replaced. The vertical white line in Part-G and the symptom described in Part-H require that the yoke assembly be replaced.

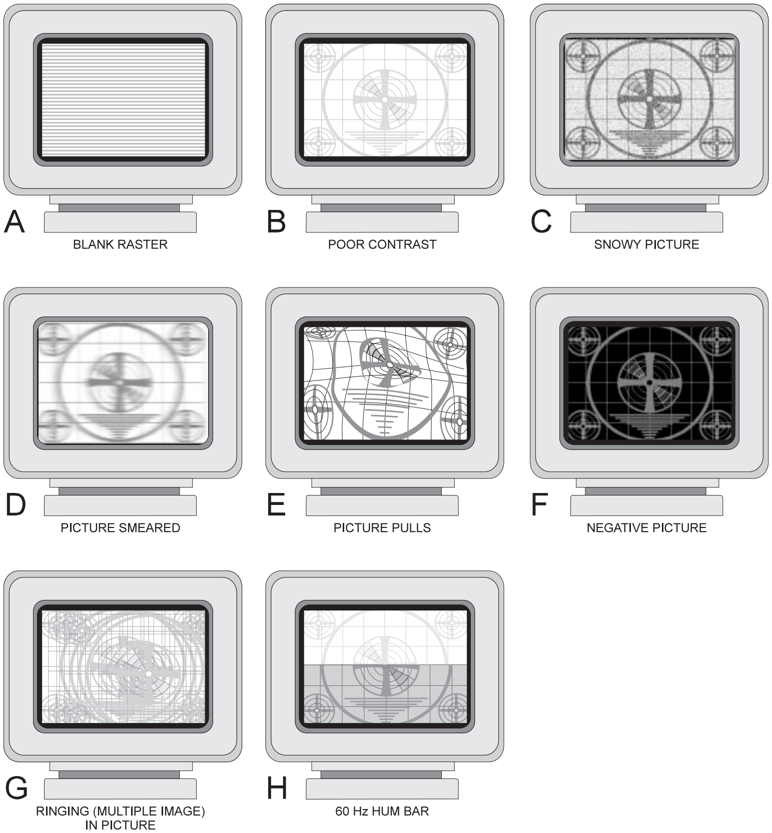
Video Problems

Objective

The symptoms depicted in Figure 9.34 represent video-signal–related problems. The blank screen depicted in Part-A indicates a complete loss of the video signal; and the pale, washed-out picture in Part-B indicates a very weak video signal. The other symptoms described (Parts C–H) are also video-signal processing problems and can usually be corrected by replacing the main board if operating and maintenance adjustments do not clear up the problem.

Figure 9.34

Video-signal–related problems.



Problems in the video signal circuitry can also produce symptoms associated with loss of vertical and horizontal sync signals.

Other Display Types

Laptop or notebook computers continue to gain popularity because of their capability to travel with the user. This has been made possible by the development of different, flat-panel display technologies. Early attempts at developing portable microcomputers used small CRTs that minimized the size of the unit. However, these units quickly gained the label of "luggables," because of their weight. The high-voltage circuitry required to operate a CRT device is heavy by nature, and could only be reduced slightly.

The most common of the flat-panel displays used for the smaller PCs are `liquid crystal displays` (LCDs). They are flat, lightweight, and require very little power to operate. In addition to reduced weight and improved portability, these displays offer better reliability and longer life than CRT units.

Liquid Crystal Displays

Objective

The liquid crystal display (LCD), illustrated in Figure 9.35, is constructed by placing `thermotropic` liquid crystal material between two sheets of glass. A set of electrodes is attached to each sheet of glass. Horizontal (row) electrodes are attached to one glass plate; vertical (column) electrodes are fitted to the other plate. These electrodes are transparent and let light pass through. A pixel is created in the liquid crystal material at each spot where a row and a column electrode intersect. A special plate called a `polarizer` is added to the outside of each glass plate. There is one polarizer on the front, and one on the back of the display.

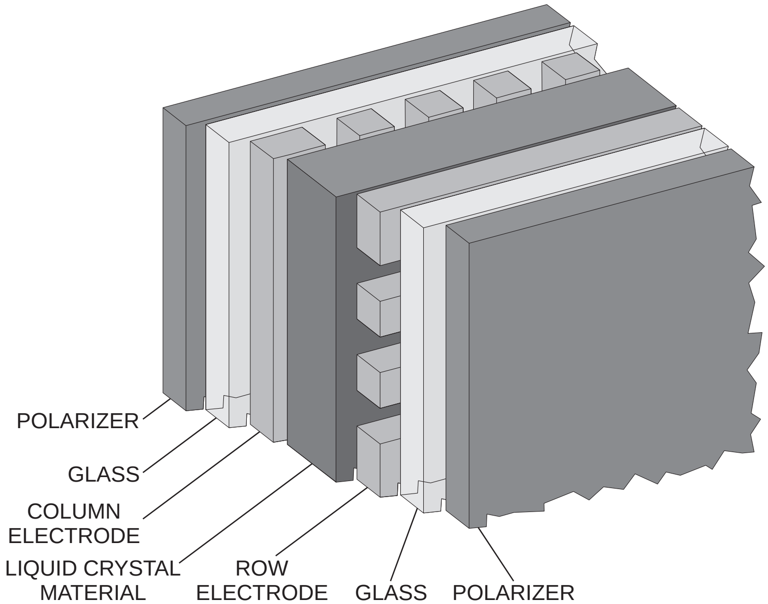


Figure 9.35

LCD construction.

The display is designed so that when the pixel is off, the molecules of the liquid crystal twist from one edge of the material to the other, as depicted in Figure 9.36. The spiral effect created by the twist polarizes light and prevents it from passing through the display. When an electric field is created between a row and column electrode, the molecules move, lining up perpendicular to the front of the display. This allows light to pass through the display, producing a single dot onscreen.

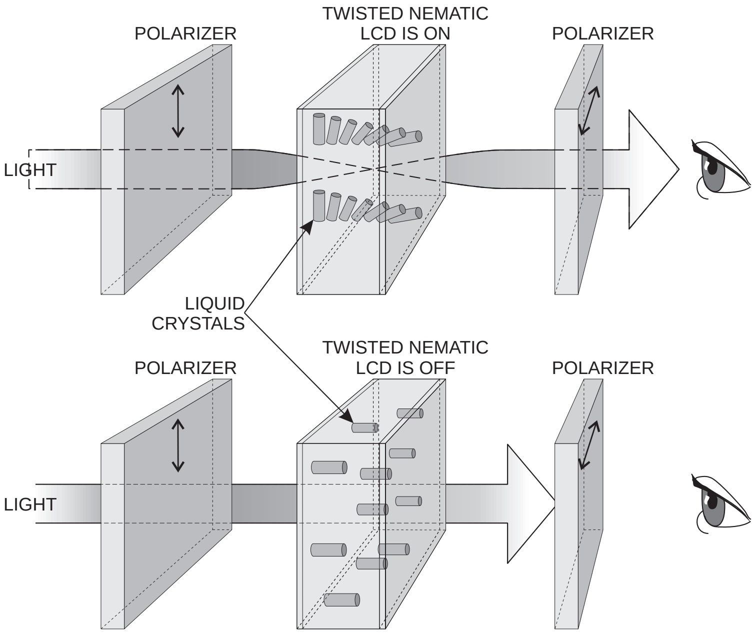


Figure 9.36

LCD operation.

Depending on the orientation of the polarizers, the energized pixels can be made to look like a dark spot on a light screen, or a light dot on a dark screen. In most notebook computers, the display is lit from behind the panel. This is referred to as `back lighting`. Some units are constructed so that the display can be removed from the body of the computer, and used with an overhead projector to display computer output on a wall or large screen.

Because no current passes through the display to light the pixels, the power consumption of LCD displays is very low. The screen is scanned using IC multiplexers and drivers to activate the panel's row and column electrodes. The row and column scanning scheme simulates the operation of the sweeping electron beam in the CRT. The scanning circuitry addresses each row sequentially, column by column. Although the column electrode is activated for a short portion of each horizontal scan, the pixels appear to be continuously lit because the scanning rate is very high. The electrodes can be controlled (turned on and off) using standard TTL levels. This translates into less control circuitry required to operate the panel.

LCD displays are available in two types, passive matrix, like the example just explained, and active matrix. The active-matrix display is similar in construction to the passive-matrix type. However, the active-matrix type adds a transistor at each row-column junction (pixel) to improve switching times. With an active-matrix display, a small current is sent to the transistor through the row-column lines. The energized transistor will conduct a larger current that is used to turn on the pixel seen on the display.

Color LCD displays are created by adding a three-color filter to the panel. Each pixel in the display corresponds to a red, blue, or green dot on the filter. Activating a pixel behind a blue dot on the filter will produce a blue dot onscreen. Like color CRT displays, the dot color onscreen of the color LCD panel is established by controlling the relative intensities of a three-dot (RGB) pixel cluster.

Gas-Plasma Displays

Objective

The gas-plasma display is similar to the LCD panel in operation. It is a gas-filled, sealed glass enclosure. The type of gas used is most typically neon, or a mixture of neon and argon. The enclosure has small, wire electrodes arranged in a row-column matrix such as the LCD panel. Where the electrodes intersect, a pixel is created. When one row and one column are energized, the gas around that intersection will discharge, giving off light and producing one dot onscreen. The voltage required for this discharge is usually less than 200 volts. The rows and columns are multiplexed and scanned to produce one full screen of information.

Summary

This chapter has investigated typical video output systems used with personal computer systems. The first portion of the chapter provided a technical description of how video-output devices work. The second section of the chapter concentrated on the various types of video adapters used with PCs. Particular attention was paid to the VGA standard adapters currently in widespread use. Both operation and troubleshooting theory was covered.

The midsection of the chapter covered the CRT monitor in some detail. Again, both operational and troubleshooting information was presented.

The final section of the chapter was used to explore different video output devices, such as LCD and gas-plasma displays.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each topic and can perform each task listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercise

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 22 - Video Problem Isolation, 23 - VGA Color Tests, 24 - Screen Attributes, and 25 - Windows 3.x Video Problems.

Review Questions

1. What portion of the system's address space is allocated to the video display adapter in an EGA system?
2. How does the operation of an interlaced and noninterlaced monitor differ?
3. What does the term `pixel` stand for?
4. The resolution of a standard VGA card is _____×_____.
5. The resolution of an SVGA card is _____×_____.
6. The resolution of an XGA card is ________×______.
7. A typical text-mode video operation produces an _____ character row by _____ column display on the monitor.
8. What video-related component should be checked first if a shaky display is observed?
9. Why should special care be taken when checking an unfamiliar video card with a 9-pin, D-shell connector?
10. List the signals associated with the output of a VGA adapter card.
11. What is the maximum resolution of a VGA card that has 256 KB of video memory installed?
12. If the video controller is operating in text mode, using a character line format of 132 characters by 25 lines, what is the minimum amount of screen memory that needs to be present on the video card?

13. How is a VGA video signal different from an MGA, CGA, and EGA signal?

14. What is the function of the shadow mask in a color CRT monitor?

15. Where should the troubleshooting sequence for any suspected video problem begin?

16. What are the three primary components likely to be involved when there is a video problem?

17. If a video problem occurs when Windows 3.x is loaded, where is the most likely place you should look for the problem?

18. What type of display paints the complete picture onscreen before a vertical retrace is performed?

19. Describe what is meant by the monitor specification .28 dot pitch. How is this measured?

20. Why should work inside the monitor's cabinet be performed very carefully by trained personnel?

21. What assumption can be made if the monitor's signal cable is removed from the adapter card and there is a raster present onscreen?

22. In an 800×600 display, what is the minimum amount of video memory required to hold two complete screens of information, if 256 colors are possible for each pixel? (8 bits = 256).

23. List the typical components found inside the monitor's cabinet.

24. Name the two types of adjustments associated with a typical computer monitor.

25. What type of problem creates a negative image onscreen?

Review Answers

1. 0A0000h through 0BFFFFh. For more information, see the section titled “Enhanced Graphics Adapter (EGA).”
2. Noninterlaced monitors present an entire image to the screen in a single sweep. Interlaced monitors produce half of the image on the first pass and the other half on a second, staggered pass. For more information, see the section titled “Creating Video Information.”
3. Picture element, or PEL. These terms are used to refer to the smallest individually definable dot or block of dots on-screen. For more information, see the section titled “Color Monitors and Graphics.”
4. 720×400 and 640×480, both using 16 colors. For more information, see the section titled “VGA Specifications.”
5. 1,280×1,024 with 256 colors. For more information, see the section titled “VGA Specifications.”
6. 1,024×768 with 265 colors. For more information, see the section titled “VGA Specifications.”
7. 25, 80. For more information, see the section titled “Storing Text Information.”
8. The video controller card. For more information, see the section titled “Hardware Checks.”
9. Because the MDA, HGA, and CGA adapters all used the same physical connector. However, the signal/pin definitions were very much different. Connecting an incompatible video adapter to a monitor could damage both the card and the monitor. For more information, see the section titled “Hardware Checks.”
10. Red video, blue video, green video, HSYNC, and VSYNC. For more information, see the section titled “Video Graphics Array (VGA) Adapters.”

11. 640×480 with 16 colors (416,000×4 bits for 16 colors = 1,664,000 bits = 208,000 bytes). For more information, see the section titled "VGA Specifications."

12. 6,600 bytes (132×25×2 for characters and attributes). For more information, see the section titled "Storing Text Information."

13. It uses analog video signals, the others use TTL-level digital signals. For more information, see the section titled "VGA Specifications."

14. It keeps the electron beam from one color gun from striking phosphor dots of the other two colors in the trio. For more information, see the section titled "Color Monitors and Graphics."

15. The first step is to check the monitor's ON/OFF switch and `Intensity` and `Contrast` controls to make certain that they are not turned off or down. The next step is to remove the video signal cable from the adapter card. If a raster appears on-screen with the signal cable removed, the problem is probably a system problem, and the monitor is good. For more information, see the section titled "Hardware Checks."

16. The monitor, the video adapter card, and the system board. For more information, see the section titled "Video Troubleshooting."

17. In the Setup window. If it is not possible to see this window on the display, just run Setup from the Windows directory in DOS. For more information, see the section titled "Windows 3.x Checks ."

18. A noninterlaced display. For more information, see the section titled "Creating Video Information."

19. Adjacent dots on the monitor screen are located 28 mm from each other. Likewise, the centers of adjacent three-color pixels would be 28 mm apart in a .28-dot pitch monitor. For more information, see the section titled "Color Monitors and Graphics."

20. There are lethal levels of voltage present inside the monitor's case. An inexperienced technician can make a deadly mistake while working in this area. For more information, see the section titled "The Monitor."

21. The monitor is functioning properly. The failure is likely to be found in the video adapter card. For more information, see the section titled "Hardware Checks."

22. 960,000 bytes. (800×600×2×8/8) For more information, see the section titled "VGA Specifications."

23. The power supply, the main signal board, the CRT tube, the deflection yoke assembly, the CRT socket board. For more information, see the section titled "Assembly/Disassembly."

24. Brightness and contrast are normal external monitor controls. For more information, see the section titled "Monitor Troubleshooting."

25. A problem with the monitor's video processing circuitry. For more information, see the section titled "Video Problems."

Chapter 10

Printers

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Describe the various methods currently used to place computer print on paper.
- Identify the major components of a dot-matrix printer.
- Discuss the types of paper handling common to different printer technologies.
- Install and configure a printer for DOS or Windows operation.
- Identify a given type of cable connection between the printer and the computer.
- Describe troubleshooting techniques associated with dot-matrix printers.
- Relate symptoms to associated components in a dot-matrix printer.
- Identify the major components of an ink-jet printer.
- Describe troubleshooting techniques associated with ink-jet printers.
- Relate symptoms to associated components in an ink-jet printer.
- Identify the major components of a laser printer.

continues

- Describe troubleshooting techniques associated with laser printers.
- Relate symptoms to associated components in a laser printer.
- Describe general alignment procedures for printhead mechanisms.
- Describe the process for applying print to a page in a laser printer.
- Exchange the FRU components of a printer (that is, power supply fuses, fans, fusers, power supplies, printheads, and so on).
- Exchange ribbon cartridges, ink cartridges, and toner cartridges in various printers.
- Disassemble and reassemble a dot-matrix printer using proper documentation.
- Disassemble and reassemble an ink-jet printer using proper documentation.

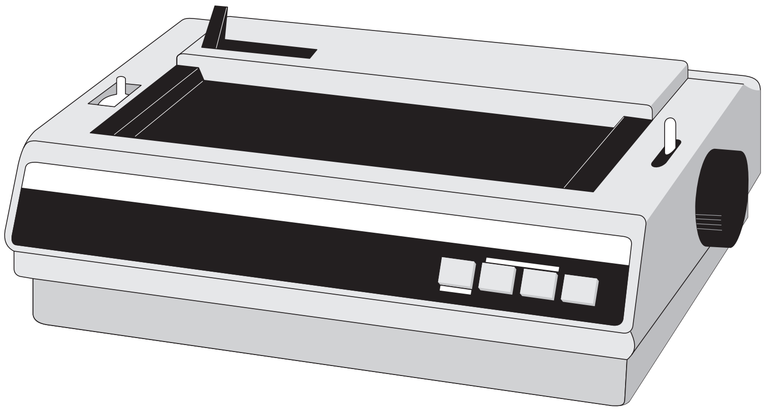
Introduction

There are many instances where a permanent copy of a computer's output may be desired. The leading hard-copy output device is the character (letters, numbers, and graphic images) printer. This definition distinguishes the character printer (generally referred to just as `printer`) from the other hard-copy output device referred to as an X-Y plotter. Plotters are typically used to create complex graphics and drawings.

Modern character printers, such as the one depicted in Figure 10.1, evolved from earlier typewriter technology. Many different mechanisms have been employed to deliver characters to a page. The earliest methods used with computer printers were just adaptations of other print mechanisms used with typewriters and teletypewriters. These included print hammers, which have characters carved on their faces, like those found in typical electric typewriters. In early computer systems, typewriters were often interfaced to the computer to provide paper copies of its output. Another adaptation from typewriter technology was the use of IBM's "golfball" printhead, borrowed from their very popular Selectric Typewriters.

Figure 10.1

A typical character printer.



Printer Characteristics

As computer systems and their applications diversified, a wide variety of printer systems were developed expressly to meet the expanding needs dictated by modern computers. Newer printing methods, such as those used in dot-matrix, ink-jet, and laser printers, have yielded much faster, higher-quality printing capabilities than ever before.

Along with the diversity of printer systems came various methods of classifying printers. Character printers may be classified by their method of placing characters on a page (`impact` or `nonimpact`), their speed of printing (`low` and `high speeds`), and the quality of the characters they produce (`fully formed`, `letter quality`, `near-letter quality`, or `dot-matrix`).

Printing Methods

Objectives

The first method of differentiating printers is by how they deliver ink to the page. Basically, the printer can produce the character by causing the print mechanism, or its ink ribbon, to impact the page. Printers that operate in this manner are referred to as impact printers.

The other printing methodology delivers ink to the page without the print mechanism making contact with the page. Printers that produce characters in this manner are known as nonimpact printers.

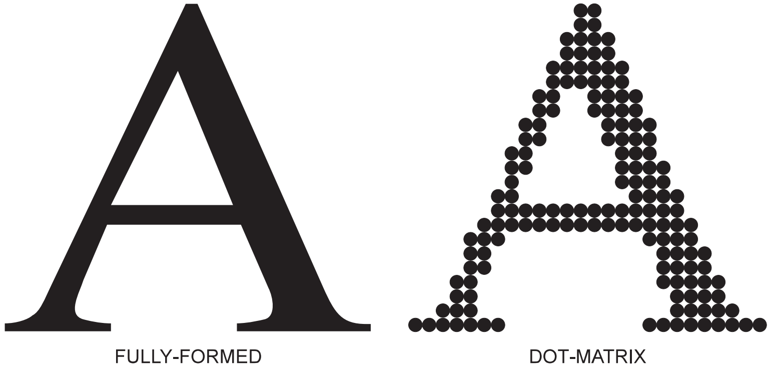
Impact Printers

Impact printers place characters on the page by causing a hammer device to strike an inked ribbon. The ribbon, in turn, strikes the printing surface (paper). The print mechanism may have the image of the character carved on its face or it may be made up of a group of small print wires, arranged in a matrix pattern. In this case, the print mechanism is used to create the character on the page by printing a pattern of dots resembling it.

Generally, the quality and therefore the readability of a fully formed character is better than that of a dot-matrix character. However, dot-matrix printers tend to be less expensive than their fully formed character counterparts. In either case, the majority of the printers in use today are of the impact variety. Figure 10.2 depicts both fully formed and dot-matrix type characters.

Figure 10.2

Fully formed and dot-matrix characters.



Nonimpact Printers

Several nonimpact methods of printing are used in computer printers. Older, nonimpact printers relied on special `heat-sensitive` or `chemically reactive paper` to form characters on the page. Newer methods of nonimpact printing use ink droplets, squirted from a jet-nozzle device, or a combination of laser/xerographic print technologies to place characters on a page.

In general, nonimpact printers are less mechanical than impact counterparts. Therefore, they tend to be more dependable. Also, to their advantage, nonimpact printers tend to be very quiet and faster than comparable impact printers. The major disadvantage of nonimpact printers is their inability to produce `carbon copies`. Nonimpact printers tend to occupy the extreme ends of the printer price range. Most of the less expensive printers are nonimpact, as are most of the very expensive, high-speed printers.

Character Types

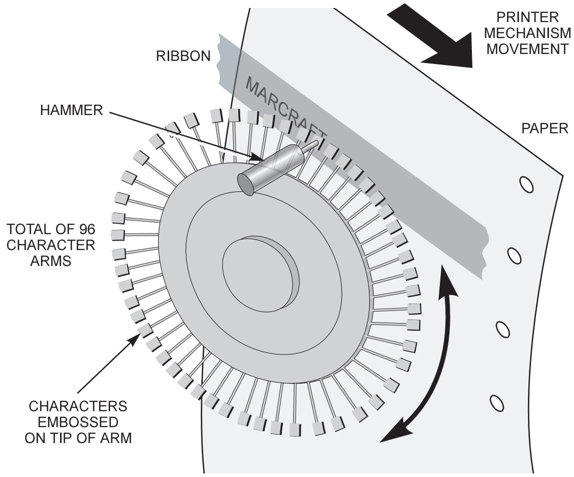
Basically, there are two methods of creating characters on a page. One method places a character on the page that is fully shaped, and fully filled-in. This type of character is called a fully formed character. The other method involves placing dots on the page in strategic patterns to fool the eye into seeing a character. This type of character is referred to as a dot-matrix character. The quality of fully formed characters is excellent. Creative choices in print fonts and sizes tend to be somewhat limited, however. To change the size or shape of a character, the print mechanism would need to be replaced. Conversely, the flexibility of using dots to create characters allows them to be altered as the document is being created. The quality of dot-matrix characters runs from extremely poor to extremely good, depending on the print mechanism.

Fully Formed Characters

The first, fully formed, impact print mechanism devised for computer printers was the `daisy wheel`, depicted in Figure 10.3. Introduced by Diablo, the daisy wheel contained an embossed character on each petal. The center hub was rotated until the correct character faced the print area. Then, a single hammer struck the petal, which struck the ribbon, which in turn struck the paper. The daisy wheel could easily be interchanged with other daisy wheels containing different fonts. The original daisy wheels were metal; newer models are plastic, making them lighter, faster, and therefore more energy efficient.

Figure 10.3

The daisy wheel.



All the fully formed, impact printing mechanisms discussed so far print one character at a time. Of the methods discussed, the daisy wheel, is by far the fastest. For higher-speed, letter-quality printing, however, characters must be printed a line at a time. This requires a line printer.

Dot-Matrix Characters

Objectives

Dot-matrix characters are not fully formed characters. Instead, dot-matrix characters are produced by printing a dot pattern representing the character, as illustrated in Figure 10.4. The reader's eye fills in the gaps between the dots. Today's dot-matrix printers offer good speed, high-quality characters that approach those created by good typewriters, and nearly limitless printing flexibility.

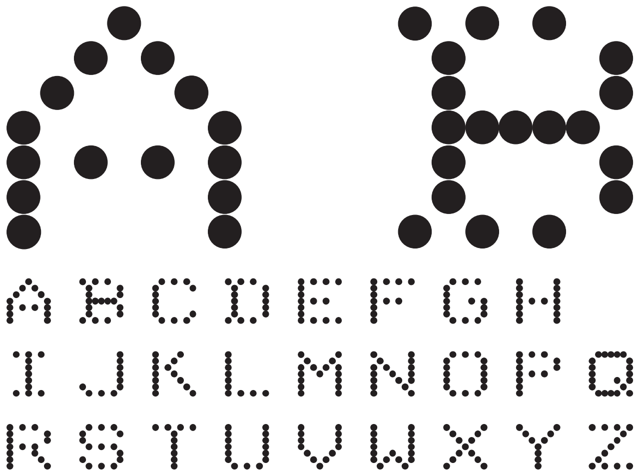


Figure 10.4

Dot-matrix characters.

Basically, the printhead in a dot-matrix printer is a vertical column of print wires that are controlled by electromagnets, as depicted in Figure 10.5. Dots are created on the paper by energizing selected electromagnets, which extend the desired print wires from the printhead. The print wires impact an ink ribbon, which impacts the paper. It is important to note that the entire character is not printed in a single instant of time. A typical printhead may contain 9, 18, or 24 print wires. The number of print wires used in the printhead is a major determining factor when discussing a printer's character quality.

The `matrix` portion of this printer's name is derived from the manner in which the page is subdivided for printing. The page is divided into a number of horizontal rows, or text lines. Each row is divided into groups of columns, called `character cells.` Character cells define the area in which a single character is printed. The size of the character cell is expressed in terms of `pitch`, or the number of characters printed per inch. Within the print cell, the matrix dimensions of the character are defined.

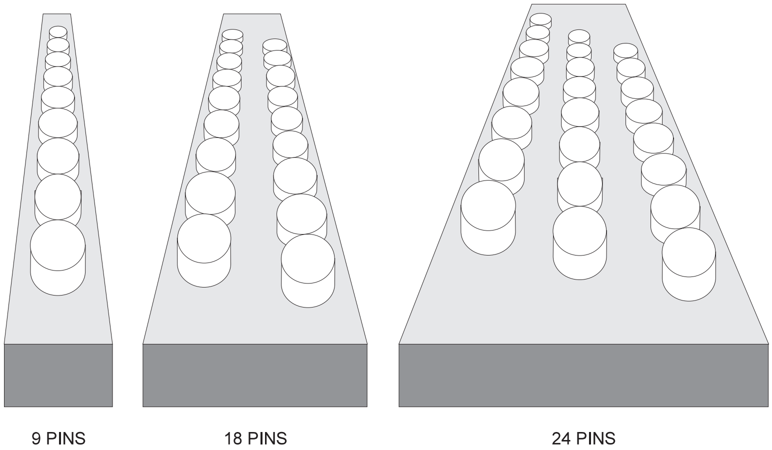


Figure 10.5

Dot-matrix printer printheads.

The density of the dots within the character cell determines the quality of the character printed. Common matrix sizes are 5×7, 24×9, and 36×18, to mention only a few of those available. The more dots that the printhead produces within the character cell, the better the character looks. This is because the dots are closer together, making the character appear more fully formed and easier to read.

Fonts

The term `font` refers to variations in the size and style of characters. With true fully formed characters, there is typically only one font available without changing the physical printing element. With all other printing methods, however, it is possible to include a wide variety of font types and sizes.

There are three common methods of generating character fonts. These are as `bitmapped` (or `raster-scanned fonts`), as `vector-based fonts`, and as `TrueType outline fonts`.

Bitmapped fonts store dot patterns for all the possible size and style variations of the characters in the set. Font styles refer to the characteristics of the font, such as `normal`, `bold`, and `italic` styles. Font size refers to the physical measurement of the character. Type is measured in increments of 1/72 of an inch. Each increment is referred to as a `point`. Common text sizes are 10- and 12-point type.

Vector-based fonts store the outlines of the character styles and sizes as sets of mathematical formulas. Each character is composed of a set of reference points and connecting lines between them. These types of fonts can be scaled up and down to achieve various sizes.

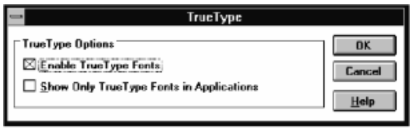
The vector-based approach requires much less storage space to store a character set and all its variations than would be necessary for an equivalent bitmapped character set. In addition, vector-based fonts can be scaled and rotated; bitmapped fonts typically cannot. Conversely, bitmapped characters can be printed out directly and quickly; vector-based characters must be generated when called for.

TrueType fonts are a newer type of outline fonts that are commonly used with Microsoft Windows. These fonts are stored as a set of points and outlines that are used to generate a set of bitmaps. Special algorithms adjust the bitmaps so that they look best at the specified resolution. After the bitmaps have been created, Windows stores them in a RAM cache that it creates. In this manner, the font is generated only one time when it is first selected. Afterwards, the fonts are just called out of memory, thus speeding up the process of delivering them to the printer.

Each TrueType character set requires an FOT and a TTF file to create all of its sizes and resolutions. Figure 10.6 depicts the TrueType enabling window under the Windows Control Panel.

Figure 10.6

TrueType enabling window.



The Control Panel's Character Map icon can be used to access the Windows Printer fonts. The Character map is depicted in Figure 10.7. This listing can also be used to insert special characters into a document.

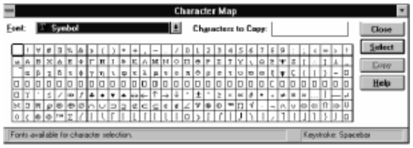


Figure 10.7

Windows character map.

Printer Speeds

The second method of classifying printers is by their speed. Low-speed printers print 300 lines per minute (10–300 characters/second); typical high-speed printers produce in excess of 20,000 lines of print per minute.

Most low-speed printers operate by printing one character at a time, across the page, in serial fashion. Therefore, these printers are commonly referred to as `serial printers`. Serial printers are usually associated with personal computers, and may use impact or nonimpact printing methods. High-speed printers generally achieve their speed by printing characters a line at a time rather than a character at a time. Therefore, they are referred to as line-at-a-time printers, or just `line printers`. Because of their cost, these printers are normally used with larger computer systems. Most line printers use impact printing methods.

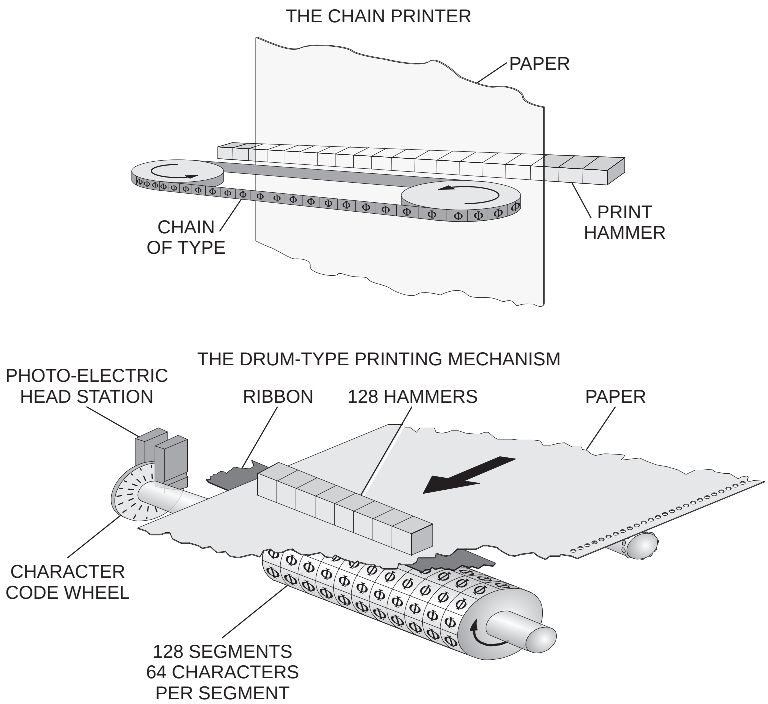
Briefly, there are two types of impact line printers:

- Rotating drum
- Chain (or band)

These printers are depicted in Figure 10.8. Rotating drum printers use a spinning, horizontal cylinder that has a complete set of characters embossed around its circumference for each character position across the page. While the drum is rotated at a constant speed, a bank of hammers strike the desired character at each print position. Actually, the hammers strike the page, which strikes the ink ribbon, which strikes the drum characters.

Figure 10.8

Rotating drum and chain printers.



Band printers have characters embossed on a flexible, metal band; chain printers place character sets on a revolving chain. In both cases, the type sets are rotated horizontally, behind the paper, at a high rate of speed. When the desired character is in the proper print position, a hammer corresponding to that position is fired, imprinting the character on the paper. Both methods use one hammer for each print position along the page. Actually, the entire line is not printed at the same instant. However, the characters are produced quickly enough to be classified as line printers. The timing logic for these printers is highly involved, because they must wait for proper character placement to occur before printing.

Laser printers deserve the title of extremely high-speed printers. These printers place lines on the page so rapidly that they virtually print a page at a time. Hence, they are called `page printers`. Laser printers in the past were very expensive and were used only with high-speed, high-volume printing operations. Recent developments in this technology, however, have produced models in a price range that makes them very attractive for use with personal computers.

Print Quality

The last criterion for comparing printers is the quality of the characters they produce. This is largely a function of how the characters are produced on the page. Printers using techniques that produce fully formed characters are described as `letter quality` (LQ) printers. All elements of the character appear to be fully connected when printed. On the other hand, those using techniques that produce characters by forming a dot pattern are just referred to as matrix printers. On close inspection of a character, one can see the dot patterns. The characters produced on some matrix printers are difficult to distinguish from those of fully formed characters. These printers have been labeled `correspondence quality` (CQ) or `near-letter quality` (NLQ) printers. Often, dot-matrix printers will have two printing modes, one being standard dot-matrix (sometimes called `utility mode`) and the other near-letter quality mode.

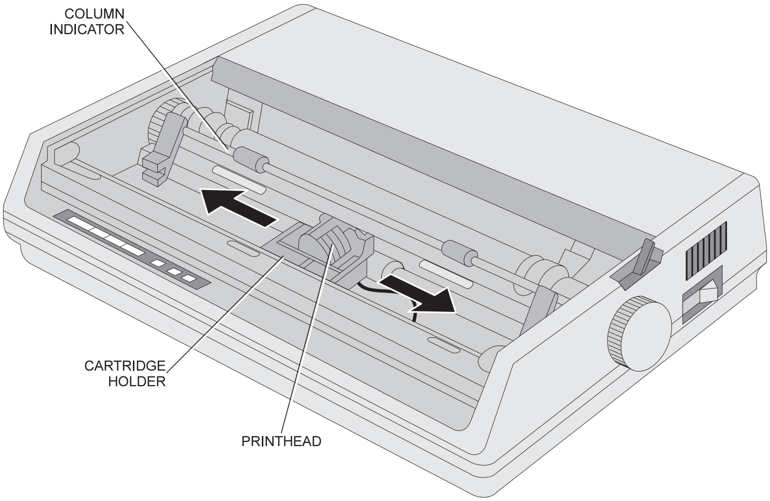
Printer Mechanics

By the very nature of their operation, printers tend to be extremely mechanical peripherals. During the printing operation, the print mechanism must be properly positioned over each character cell, in sequence. Loss of synchronization in contact printers can lead to paper jams, tearing, smudged characters, or printhead damage. Noncontact printers may produce totally illegible characters, if synchronization is lost. The positioning action may be produced by moving the paper under a stationary printhead assembly or by holding the paper stationary and stepping the printhead carriage across the page. In the last operation, the `printhead carriage` rides on rods extending across the front of the page, as shown in Figure 10.9.

Depending on the type of print mechanism used, the carriage may be stepped across the page at a rate of one character cell at a time (fully formed characters), or in sub-character-cell steps (dot-matrix characters). Printing may occur in only one direction (unidirectional) or in both directions (bidirectional.) In bidirectional printers, the second line of characters is stored in the printer's buffer memory and printed in the opposite direction, saving the time that would normally be used to return the carriage to the start of the second line.

Figure 10.9

The printhead carriage.



The printhead carriage assembly is stepped across the page by a `carriage motor/timing belt` arrangement. With many printer models, the number of character columns across the page is selectable, producing variable characters spacing (expressed in `characters per inch`, or CPI) that must be controlled by the carriage-position motor. Dot-matrix printers may also incorporate variable dot densities (expressed as `dot-pitches`). Dot-pitch is also a function of the carriage motor control circuitry. Obviously, this discussion excludes continuous-stream, ink-jet printers, where printing is done by electromagnetic deflection of the ink drops, and laser printers, where the beam is reflected by a revolving mirror.

Paper Handling

Objectives

In addition to positioning the print mechanism for printing, all printer types must feed paper through the print area. The type of `paper handling mechanism` in a printer depends somewhat on the type of form intended to be used with the printer, and its speed. Paper forms fall into two general categories, `continuous forms`, which come in folded stacks and have holes along their edges, and `single-sheet forms`, such as common typing paper.

There are two common methods of moving paper through the printer:

- Friction-feed—Uses friction to hold the paper against the printer's `platen`. The paper advances through the printer as the platen turns.
- Pin-feed—Pulls the paper through the printer by a set of pins that fit into the holes along the edge of the form, as shown in Figure 10.10. The pins may be an integral part of the platen, or mounted on a separate, motor-driven `tractor`.

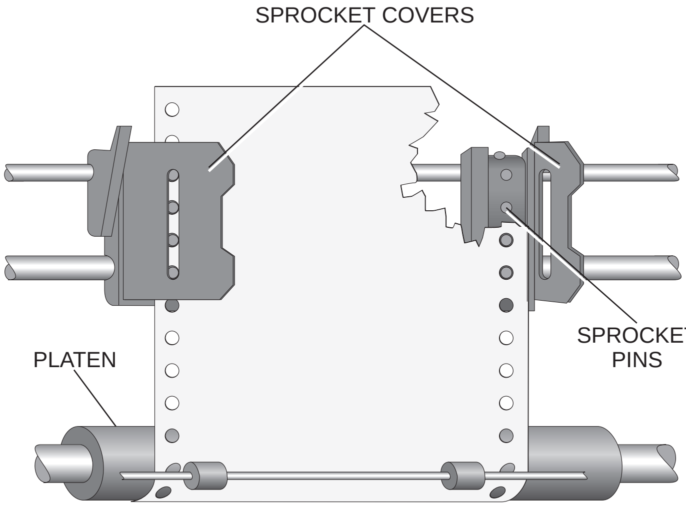


Figure 10.10

A pin-feed tractor mechanism.

Friction feed is normally associated with single-sheet printers. The sheet-feeding system can be manual or automatic. Platen pin-feed and pin tractors are usually employed with continuous and multi-layer forms. These mechanisms can control paper slippage and misalignment created by the extra weight imposed by continuous forms. Platen pin-feed units can handle only one width of paper; tractors can be adjusted to handle various paper widths. Tractor feeds are used with very heavy forms, such as multiple part, continuous forms, and are most commonly found on dot-matrix printers. Most ink-jet and laser printers use single-sheet feeder systems.

The gear trains involved in the paper handling function can be treated as an FRU item in some printers. While it is possible to replace the gears, or gear packs, in dot-matrix and ink-jet printers (if they can be obtained from the manufacturer as separate items), it is not usually economical to do so. Laser printers on the other hand, are normally expensive enough to warrant replacing the gear trains and clutch assemblies that handle the paper movement through the printer.

Printer Controls

Although printers vary considerably from type to type, and model to model, some elements are common to all printers. These elements are depicted in Figure 10.11.

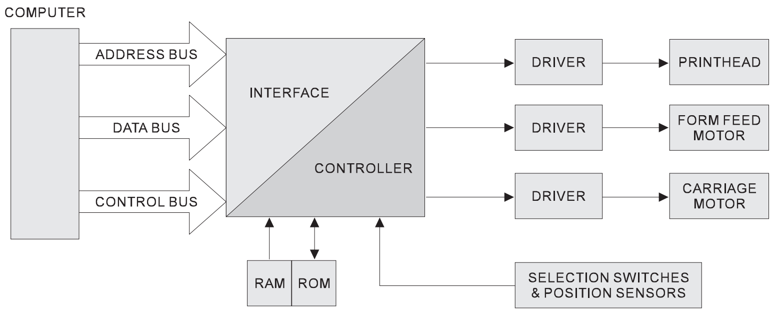


Figure 10.11

Common printer components.

Like most other peripherals, the heart of a character printer is its `interface/controller` circuitry. The interface circuitry accepts data and instructions from the computer's bus systems and provides the necessary interfacing (serial or parallel) between the computer and the printer's control circuitry. This includes decoding the computer's instructions to the printer, converting signal-logic levels between the two, and passing data to the printer's controller.

Parallel port connections are most efficient when the printer is located in close proximity to the computer. If the printer must be located remotely, the serial interface becomes more appropriate. Many manufacturers offer both connections as standard equipment. Others offer the serial connection as an option. More is said about these two interfaces later in this section. A third, less common, method of connecting printers to computers uses the SCSI interface as the connection port. As with other SCSI devices, the printer must be set up as a unique SCSI device and observe proper connection and termination procedures.

The controller section receives the data and control signals from the interface section and produces all the signals necessary to select, or generate, the proper character to be printed. It also advances the print mechanism to the next print position and feeds the paper at the proper times. In addition, the controller generates status and control signals that tell the computer what is happening in the printer.

Because of the complexity of most character printers, a dedicated `microcontroller` is commonly used to oversee the operation of the printer. The presence of the onboard microprocessor provides greater flexibility and additional options for the printer.

Along with the dedicated processor, the printer normally contains onboard memory in the form of RAM, ROM, or both. A speed mismatch exists between the computer and the printer because the computer is normally capable of generating characters at a much higher rate than the printer can print them. To minimize this speed differential, printers typically carry onboard-RAM memory buffers to hold characters coming from the computer. In this way, the transfer of data between the computer and the printer occurs at a rate compatible with the computer's operating speed. The printer obtains its character information from the onboard buffer.

In addition to character print information, the host computer can also store printer instructions in the buffer for use by the dedicated processor. The printer may also contain `onboard ROM` in the form of `character generators`, or `printer initialization` programs for start-up. Some printers contain EPROM, rather than ROM, to provide a greater variety of options for the printer, such as downloadable type fonts and variable print modes.

In some printers, the microcontroller, RAM chips or modules, and the ROM/EPROM devices may be treated as FRU components. Many laser printers come with a preset amount of RAM on board, but allow the memory to be upgraded if needed. Many high speed laser printers require additional RAM to be installed to handle printing of complex documents, such as desktop published documents containing large *Encapsulated Post Script* (EPS) graphics files. Similarly, ROM and EPROM devices that contain BIOS or character sets are often socketed, so that they can be replaced or upgraded easily.

As with the gears and gear trains discussed earlier in the chapter, the replaceability of these units depend on the ability to source them from a supplier. In most cases, the question is not one of "Can the device be exchanged?", its one of whether it makes economical sense to do so. For a given printer type and model, the manufacturer's service center can provide information about the availability of replacement parts.

Basically, the controller must produce signals to drive the print mechanism, the paper-feed motor, the carriage motor, and possibly optional devices such as single-sheet feeders and add-on tractors. Most of these functions are actually performed by precision stepper motors. There are usually hardware driver circuits between the motors and the controller to provide current levels high enough to activate the motors.

The controller also gathers information from the printer through a variety of sensing devices. These include position sensing switches, and user-operated, front-panel—mounted, mode-control switches. Some of the more common sensing switches include the `home-position` sensor, `end-of-paper` sensor, and the `carriage-position` sensor. The controller also responds to manual input command switches such as `On/Offline`, `Form Feed` (FF), and `Line Feed` (LF).

Printer Installation

Objectives

Generally speaking, one of the least difficult I/O devices to add to a microcomputer system is a parallel printer. This is largely because from the beginning of the PC era, a parallel printer has been one of the most standard pieces of equipment to add to the system. This standardization has led to fairly direct installation procedures for most printers. Obtain an IBM Centronics printer cable, plug it into the appropriate LPT port on the back of the computer, connect the Centronic-compatible end to the printer, plug the power cord into the printer, load a device driver to configure the software for the correct printer, and print.

Serial printers are slightly more difficult to set up because the communication definition must be configured between the computer and the printer. The serial port will need to be configured for speed, parity type, character frame, and protocol.

Regardless of the type of printer being installed, the steps for adding a printer to a system are basically the same. Connect the printer to the correct I/O port at the computer system. Make sure the port is enabled. Set up the appropriate printer drivers. Configure the port's communication parameters, if a serial printer is being installed. Install the paper. Run the printer's self-test, and then, print a document. These steps are summarized in Figure 10.12.

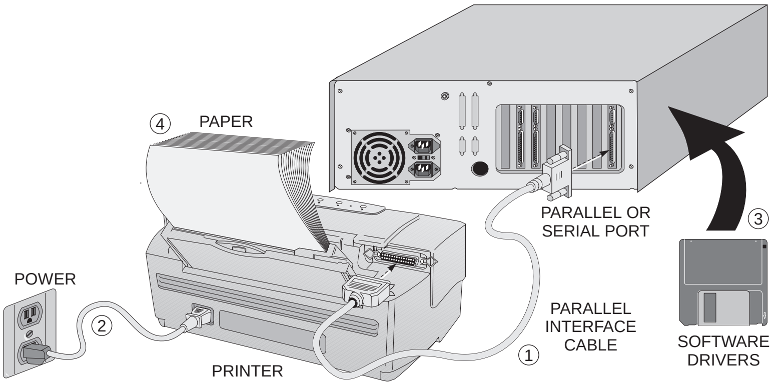


Figure 10.12

Printer installation steps.

Printer Drivers

Like mice, printers require device driver programs to oversee their operation. A software developer who is writing a word processing program, for example, will not have any way of knowing what type of printers will be used. Although most printers use the same codes for alphanumeric data, they may use widely different control and special-feature codes to produce special text and to govern the printer's operation. Therefore, software producers often develop the core of a program, and then offer a disk full of printer drivers to translate between the software package and different standard printers. The user normally selects the driver program needed to operate the system through a configuration program that comes with the software. This function is usually performed the first time the software is loaded into the system. Figure 10.13 illustrates the functional position of a printer driver in the system.

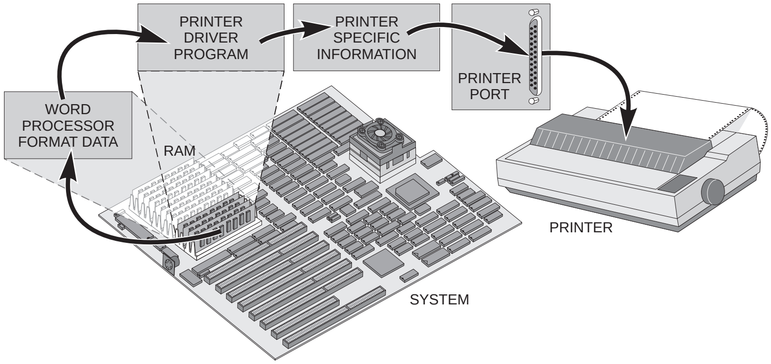


Figure 10.13

Printer driver position.

Driver programs may be supplied by the software developer as part of the package, or by the hardware developer. It is often in the best interests of a hardware developer to offer drivers that make the hardware compatible with popular pieces of application software. Conversely, if a software developer is introducing a new piece of software, they often offer drivers that make the software compatible with as many hardware variants as possible.

Serial Printer Considerations

Objective

In some applications, it is just impossible to locate the printer close enough to the host computer to use the parallel connection. In these cases, serially interfaced printers come into play. Printers using a serial interface connection add an extra level of complexity to the system. Unlike the parallel-printer interface that basically plugs and plays on most systems, serial interface connections require additional hardware and software configuration steps. Serial printer problems basically fall into the following three categories:

- Cabling problems
- Configuration problems
- Printer problems

Cabling Problems

Not all serial cables are created equal. In the PC world, RS-232 serial cables can take on several configurations. First of all, they may use either 9-pin or 25-pin, D-shell connectors. The cable for a particular serial connection will need to have the correct type of connector at each end. Likewise, the connection scheme inside the cable can vary from printer to printer. Normally, the transmit data line (`TXD` - pin 2) from the computer is connected to the receive data line (`RXD` - pin 3) of the printer. Also, the data set ready (`DSR` - pin 6) is typically connected to the printer's data terminal ready (`DTR` - pin 20) pin. These connections are used as one method to control the flow of information between the system and the printer. If the printer's character buffer becomes full, it will signal the computer to hold up sending characters by deactivating this line.

Different or additional pin interconnections can be required for other printer models. The actual implementation of the RS-232 connection is solely up to the printer manufacturer. Figure 10.14 depicts typical connection schemes for both 9-pin and 25-pin connections to a typical printer. The connection scheme for a given serial-printer model is normally provided in its User's Manual.

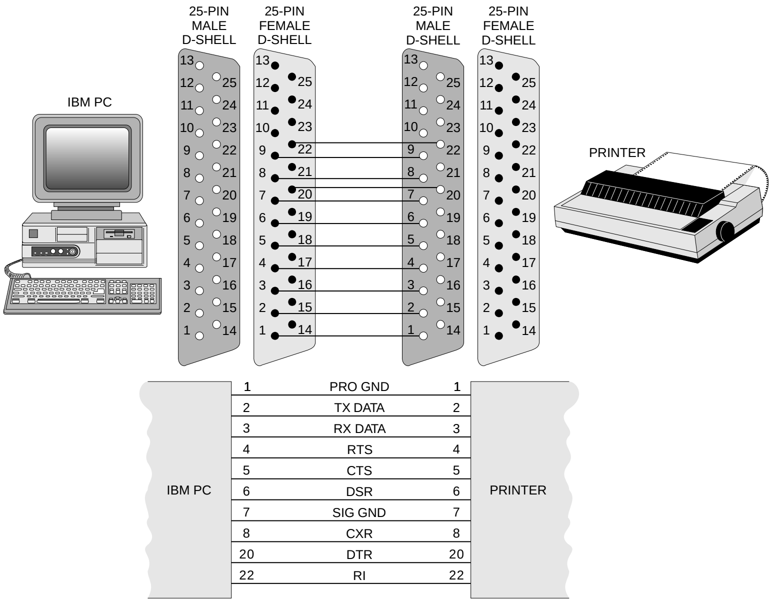


Figure 10.14

Serial-printer–connection schemes.

Configuration Problems

After the correct connector and cabling scheme has been implemented, the printer configuration must be established at both the computer and at the printer. The information in both locations must match for communications to go on. On the system side of the serial port connection, the software printer driver must be set up to match the setting of the printer's receiving section.

First, the driver must be directed toward the correct serial port. In a Windows-based system, this is typically COM2. Second, the selected serial port must be configured for the proper character framing. The number of start, stop, data, and parity bits must be set to match what the printer is expecting to receive. These values are often established through hardware configuration switches located on the printer.

The printer driver must also be set up to correctly handle the flow of data between the system and the printer. Incorrect flow control settings can result in slow response, lost characters, or continuous errors and retries. Flow control can be established through software or hardware handshaking. In a hardware-handshaking mode, the printer tells the port that it is not prepared to receive data by deactivating a control line, such as the DTR line. Conversely, in a software-handshaking environment, control codes are sent back and forth between the printer and the computer to enable and disable data flow.

Two popular methods of implementing software flow control are XON/XOFF and ETX/ACK. In the XON/XOFF method, special ASCII control characters are exchanged between the printer and the computer to turn the data flow on and off. In an ETX/ACK protocol, ASCII characters for end-of-text (ETX) and acknowledge (ACK) are used to control the movement of data from the port to the printer. Basically, the computer attaches the ETX character to the end of a data transmission. When the printer receives the ETX character, it checks the incoming data, and when ready, returns an ACK character to the port. This notifies the system that the printer is capable of receiving additional characters. This concept is illustrated in Figure 10.15. In any event, both ends of the interface connection must be set to use the same flow-control method.

Serial communications standards and procedures are covered in greater detail in Chapter 11, "Data Communications." Consult this information for more information about character framing, error detection and correction methods, and serial transmission protocols.

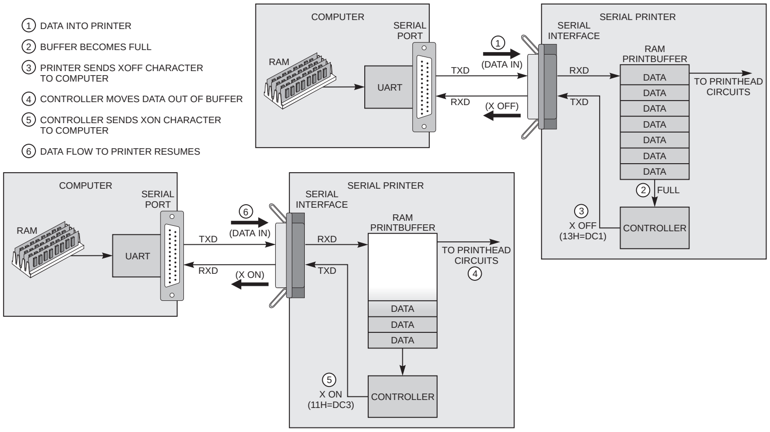


Figure 10.15

Software flow control.

Printer Problems

Problems associated with serial printers differ from those of parallel printer only in the area of the serial interface configuration. As mentioned in the preceding "Configuration Problems" section, the protocol, character framing, and baud rate of the printer must match that of the system's interface. After ensuring that the interface settings match and that the interface is working, the steps of troubleshooting a serial printer are identical to those given for parallel interfaced printers. Therefore, the only steps that need to be added to the troubleshooting sections in the later sections of this chapter are those needed to validate the operation of the serial interface.

Installing Printers in Windows 3.x

To install printers through Windows 3.x, double-click on the Printers icon in the Control Panel. This activates the Printers dialog box. The dialog box is divided into three separate sections labeled Default Printer, Installed Printers, and List of Printers. When Windows is first installed, no printers are listed in the Default Printer or Installed Printers sections. To install a printer, select the Add option, which will cause the List of Printers to appear. The printer to be installed is selected from this list. The specific brand and model of the printer can be obtained from the printer itself. Then when a match is found and selected, click on the Install option.

Windows will search the A: drive for an appropriate printer driver for the selected printer. None of the drivers are included at the time Windows is installed because most of them will not be needed and they consume hard-disk drive space. After Windows loads the driver, the printer will appear in the Installed Printers section. Figure 10.16 depicts the Windows 3.x Printer Installation window.

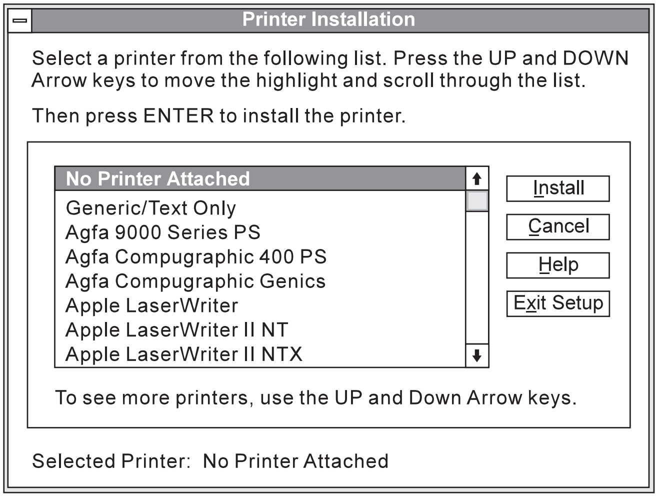


Figure 10.16

Windows 3.11 Printer Installation window.

Another common solution is to contact the manufacturer of the printer for an appropriate driver. Most of the time, the driver is included when the printer is purchased. When installing a printer that is not listed, select Unlisted Printer from the List of Printers section, and select the Install option. Windows then searches the A: drive for the OEM driver software.

After all the drivers have been installed, the desired printer must be selected from the Installed Printers section, using the Set as Default Printer option. The printer will appear in the Default Printer section.

Printers in Windows 95

Printing is significantly improved in Windows 95. The Print Manager function and its support components have been integrated into a single print processing architecture referred to as the *Print Spooler*. This integration provides smooth printing in a background mode and quick return-to-application time. The key to this operation is in how the print spooler sends data to the printer. Data is moved to the new printer only when it is ready. Therefore, the system is never waiting for the printer to digest data that has been sent to it.

Windows 95 automatically adopts any printers that have been established before its installation. If no printers are already installed, the Setup program will run the new Add Printer wizard to allow a printer to be installed. Each printer in the system has its own print window and icon to work from. The wizard can be accessed at any time through the Windows 95 Desktop and Start menu. In the Start menu, move to Settings and click Printers. Likewise, through the My Computer icon, or the Control Panel window, double-click the Printers folder or icon.

To install a printer, open the Printers folder and double-click the Add Printers icon. From this point, the Printer wizard guides the installation process. Because Windows 95 has built-in networking support, the printer can be a local unit (connected to the computer), or a remote unit located somewhere on the network.

If the printer is connected to a remote computer (print server), the remote unit must supply the printer drivers and settings to control the printer. Likewise, the print server must be set up to *Share* the printer with the other users on the network. To install the network printer, access the Network Neighborhood icon on the Desktop, select the remote computer's network name, the remote printer's name, and right-click the Install option. After the remote printer has been installed, the local computer can access it through the Network Neighborhood icon.

If the printer is not recognized as a model supported by the Windows 95 driver list, OEM drivers can be installed from a disk containing the OEMSETUP.INF file.

To print an open file in Windows 95, move to the File menu as normal and click the Print option. If the file is not open, it is still possible to print it by dragging its icon onto the desired printer's icon.

To view documents waiting to be printed from the Print Spooler, double-click the desired printer's icon. This will display the existing print queue. Unlike the Windows 3.x Print Manager, closing the print window does not interrupt the print queue in Windows

95. The print jobs in the queue will be completed unless the jobs are deleted.

Dot-Matrix Printers

The stalwarts of microcomputer printing have been the dot-matrix impact printers. They consist of a power-supply board, a main control board, a printhead assembly, a ribbon cartridge, a paper-feed motor (along with its mechanical drive gears), and a printhead-positioning motor and mechanisms.

The Power Supply

The power-supply board is called on to provide various voltages to power the electronics on the control board. It also drives both the printhead-positioning and paper-feed motors, and energizes the wires of the printhead so that they will strike the ribbon as directed by the control board.

The Main Control Board

The control board is typically divided into four functional sections, as described in Figure 10.17. These functional blocks include the following:

- The interface circuitry
- The character generation circuitry
- The printer controller circuitry
- The motor control circuitry

The control board contains the logic circuitry required to convert the signals received from the computer's adapter card into character patterns, as well as to generate the proper control signals to position the printhead properly on the page, fire the correct combination of printhead wires to create the character, and to advance the paper properly. The onboard microcontroller, character generators, RAM, and ROM are found on the control board.

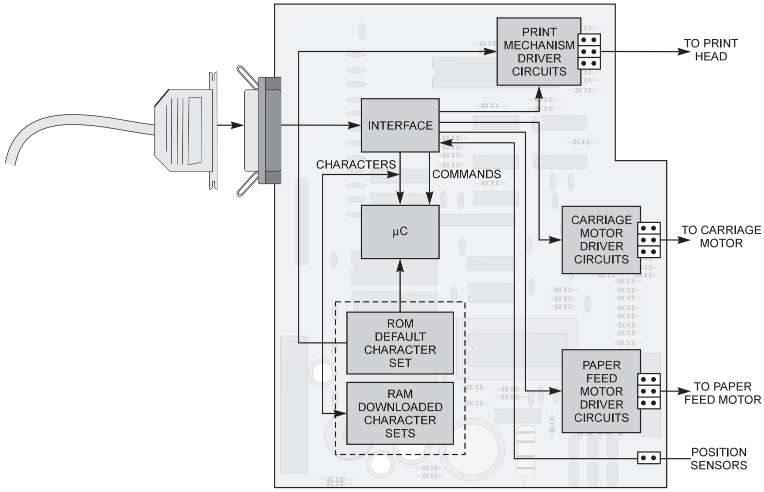


Figure 10.17

Logical parts of the control board.

The status of the printer's operation is monitored by the control board through a number of sensors. These sensors typically include the following:

- Paper out
- Printhead position
- Home position (for the printhead carriage)

Input from the printer's operator panel is also routed to the control board. Operator panel information includes the following:

- Online
- Form feed
- Line feed
- Power/Paper out

The Control panel may contain a number of other buttons and indicator lights, whose functions are specific to that particular printer. Always consult the printer's User's Manual for information about the Control panel buttons and indicators.

The printer's interface may contain circuitry to handle serial data, parallel data, or a combination of the two interface types. At the printer end of a Centronics parallel port, a 36-pin connector, like the one depicted in Figure 10.18, is used.

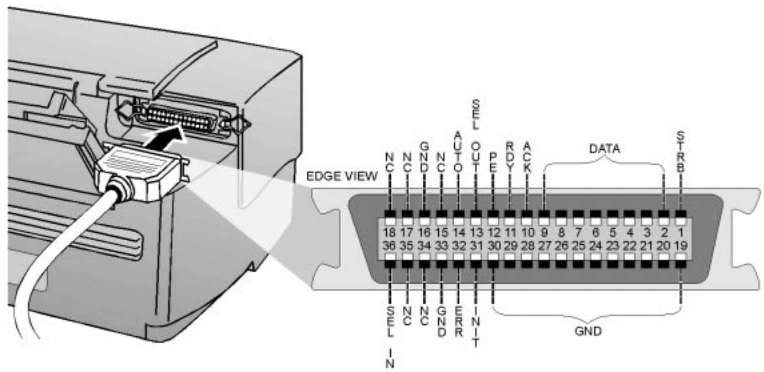


Figure 10.18

A parallel connection at the printer.

Dot-matrix printers process bit patterns in much the same way that CRT controllers do. The dot patterns are accessed from a character-generator ROM. In addition to the standard ASCII character set, many printers feature preprogrammed sets of block-graphics characters that can be used to create nontext images on a page. Most manufacturers use EPROM (erasable-programmable ROM) character generators rather than the older, ROM type. This allows their units to accept downloadable fonts from software. Used with a high-quality printhead, a variety of typefaces such as Roman Gothic, Italic, and foreign language characters can be loaded into the programmable character generator from software. In addition, it is possible for the user to create his own character sets, typefaces, and graphic symbols. Some manufacturers even offer standard, bar-code graphics software sets for their machines.

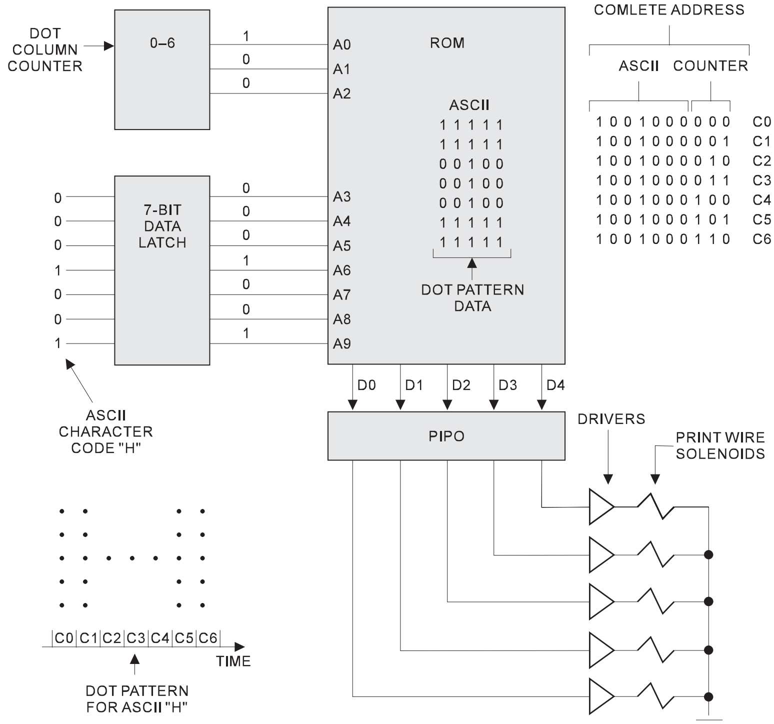
Printhead Mechanisms

The `printhead` is a collection of print wires set in an electromagnetic head unit. The printhead assembly is made up of a permanent magnet, a group of electromagnets, and a housing. In the printhead, the permanent magnet keeps the wires pulled in until electromagnets are energized, causing them to move forward. The printhead is mounted in the `printhead carriage assembly`. The carriage assembly rides on a bar that passes along the front of the platen. The printhead carriage assembly is attached to the printhead-positioning motor by a `timing belt`.

Figure 10.19 illustrates a typical dot-matrix, printhead control circuit. Although the character generator may appear very similar to the one used with CRT controllers, the data must be processed differently because the vertical nature of the printhead mechanism, as opposed to the horizontal nature of the CRT's scan lines. In this case, the portion of the character generator containing the dot pattern for an uppercase H is shown. The ASCII dot-column code required to access it is also presented. Each time the dot-column counter is pulsed, the address is incremented, the print wires are activated, and the printhead carriage is stepped over one place in the character cell.

Figure 10.19

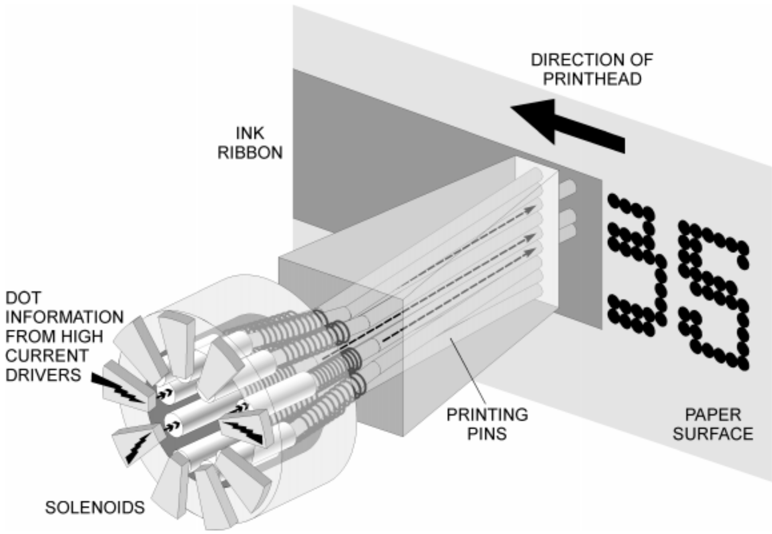
A typical dot-matrix printhead control circuit.



The printhead-positioning motor is responsible for moving the printhead mechanism across the page and stopping it in just the right places to print. The printhead rides back and forth across the printer on a pair of carriage rods. A timing belt runs between the printhead assembly and the printhead-positioning motor, and converts the rotation of the motor into linear movement of the printhead assembly. The printhead must stop each time the print wires strike the paper. If this timing is off, the characters will be smeared on the page and the paper may be damaged. The motor steps a predetermined number of steps to create a character within a character cell. Figure 10.20 illustrates a dot-matrix printhead delivering print to a page.

Figure 10.20

Dot-matrix printhead.



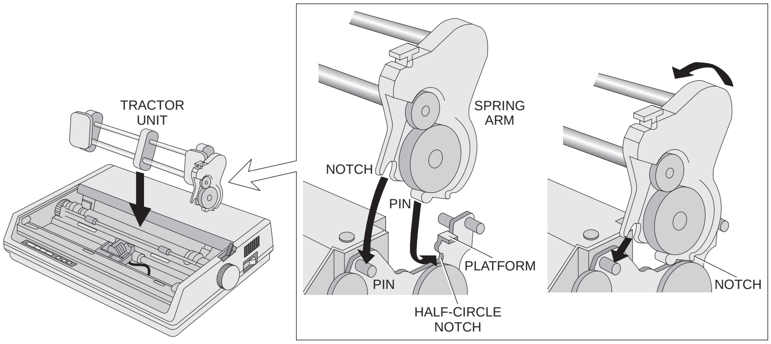
Paper Handling

The paper-feed motor and gear train move the paper through the printer. This can be accomplished by driving the platen assembly. The platen can be used in two different ways to move the paper through the printer. After the paper has been wrapped halfway around the platen, a set of rollers are used to pin the paper to the platen as it turns. This is `friction-feed paper handling`. As described earlier, the platen may have small pins that can drag the paper through the printer as the platen turns. In either case, the paper-feed motor drives the platen to move the paper.

The feed motor's gear train can also be used to drive the extended gear train of a tractor assembly, when it is installed. The gears of the feed motor mesh with those of the tractor, causing it to pull or push the paper through the printer. To use a tractor, the friction-feed feature of the platen must be released. Otherwise, the tractor and the platen may not turn at the same rate and the paper will rip or jam. The installation of a tractor assembly is illustrated in Figure 10.21.

Figure 10.21

Installing a tractor assembly.



Objectives

Color Printing

Another interesting innovation available with dot-matrix printers is `color printing`. Color printing capability can be divided into two parts: `dumb colors`, obtained by shifting a multicolor ribbon to the correct level; or `smart colors`, which are created by using multiple print passes to interlace dumb colors from the ribbon (double-strike mode).

Troubleshooting Dot-Matrix Printers

The classical first step in determining the cause of any printer problem is to determine which part of the printer-related system is at fault—the computer, the cable, or the printer. Nearly every printer is equipped with a built-in self-test. The easiest way to determine whether the printer is at fault is to run its self-test. Consult the printer's User's Manual for instructions in running its self-test. Some printers are capable of producing audible tones to indicate the nature of an internal problem. Refer to the printer's User's Manual for the definitions of the coded, beep tones, if they are available.

If the printer runs the self-test and prints clean pages, most of the printer has been eliminated as a possible cause of problems. The problem could be in the computer, the cabling, or the interface portion of the printer. If the printer fails the self-test, however, it will be necessary to diagnose the printer's problem. The following section presents typical problems encountered in dot-matrix printers.

The following are symptoms for dot-matrix printer problems:

- No lights or noise from printer.
- Light or uneven print being produced.

- Printhead moving, but not printing.
- Dots missing from characters.
- Printhead printing, but does not move.
- Paper will not advance.

Dot-Matrix Printer Configuration Checks

The presence of onboard microcontrollers allows modern printers to be very flexible. Like other peripheral devices, printers can be configured to operate in different modes. Operating configuration information can be stored in CMOS RAM on the control board. Some configuration settings may be made through DIP switches mounted inside the printer. These switches are read by the printer's microcontroller at startup.

In the case of dot-matrix printers, the configuration settings are normally entered into the printer through the buttons of its Control Panel. Typical dot-matrix configuration information includes the following:

- Printer mode
- Perforation skip (for continuous forms)
- Automatic line feed at the bottom of the page
- Paper handling type
- ASCII character codes (7-bit or 8-bit)
- Basic character sets

Other quantities that can be set up include the following:

- Print font
- Character pitch
- Form length

Most dot-matrix printers contain two or three onboard fonts (character styles) that can be selected through the printer's configuration routines. Typical fonts included in dot-matrix printers are as follows:

- Draft
- Courier
- Prestige
- Bold prestige

In many dot-matrix printer models, it is also possible to download other fonts from the computer. The character pitch refers to the number of characters printed per inch. Common pitch settings include 10, 11, 12, and 14 dots per inch (dpi). Consult the printer's User's Guide to find the definitions of such settings.

Dot-Matrix Printer Hardware Checks

To perform work inside the printer, it is necessary to disassemble its case. Begin by removing any add-on pieces such as dust covers and paper feeders. Next, remove the paper advancement knob located on the right side of most dot-matrix printers. Turn the printer over and remove the screws that hold the two halves of the case together. These screws are sometimes hidden beneath rubber feet and compliance stickers. Finally, it may be necessary to disconnect the printer's front panel connections from the main board to complete the separation of the two case halves. This procedure is described in Figure 10.22.

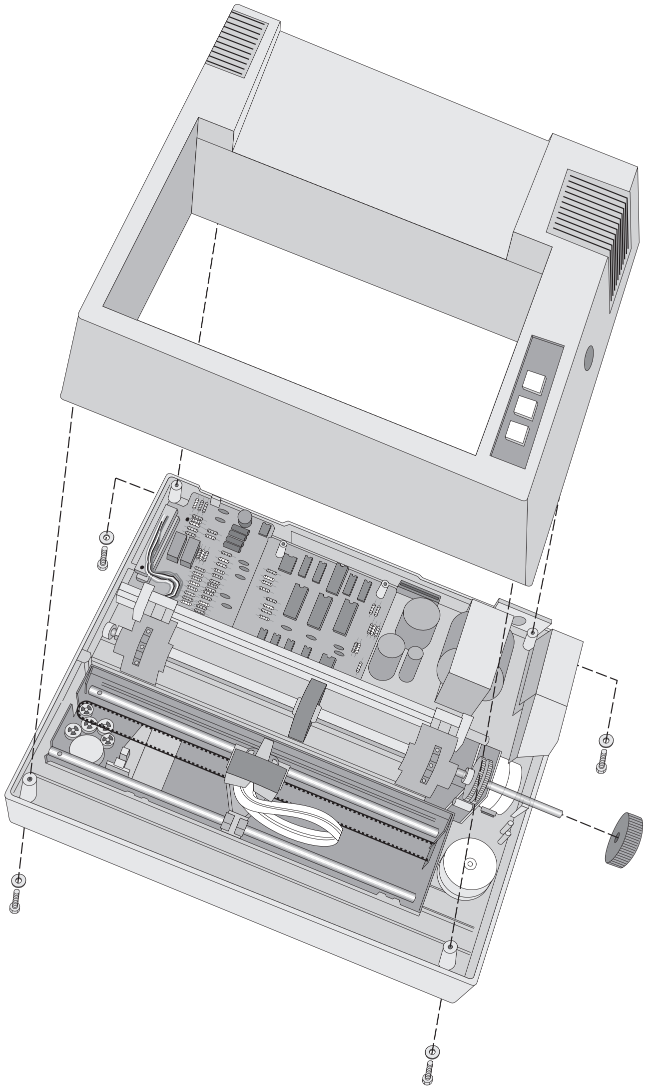
Dot-Matrix Printer Power-Supply Problems

If the printer will not function and displays no lights, no sounds, and no actions, the power supply is generally involved. Troubleshoot printer power-supply problems in the same manner as a computer power supply. As a matter of fact, the power-supply troubleshooting routine is the same.

Check the Online light. If the printer is offline, no print action will occur. A missing or improperly installed ribbon cartridge will also prevent the unit from printing. Install the ribbon correctly. Check the power outlet to make certain that it is live. Plug a lamp or other device in the outlet to verify that it is operative. Check to see that the power cord is plugged in securely to the printer and the socket. Make sure the power switch is turned on.

Figure 10.22

Disassembling the printer.



If everything is plugged in and in the on position, but still not working, turn off the power and unplug the printer from the outlet. Remove the top of the printer's case and find the power-supply board. Check the power-supply's fuse to make sure that it is good. If the fuse is blown, replace it with a fuse of the same type and rating. Do not replace a blown fuse with a conductor or a slow-blow fuse. To do so could lead to more extensive damage to the printer, and possible unsafe conditions.

Also, check the power supply and control boards, as well as the paper-feed and printhead-positioning motors for burnt components or signs of defect. Fuses do not usually blow unless another component fails. The other possible cause of excessive current occurs when a motor (or its gear train) binds and cannot move. Check the drive mechanisms and motors for signs of binding. If the gear train or positioning mechanisms will not move, they may need to be adjusted before replacing the fuse.

If none of the printer sections work when everything is connected and power is applied, it will be necessary to exchange the power-supply board for a new unit. Unlike the computer's power supply, the typical printer power supply is not enclosed in a protective housing and, therefore, presents a shock hazard anytime it is exposed.

To exchange the power-supply board, disconnect the power cable from the printer. Disconnect and mark the cabling from the control board and any other components directly connected to the power supply. Remove any screws or clips that secure the power-supply board to the case. Lift the board out of the cabinet. Install the new board and reconnect the various wire bundles to it.

Ribbon Cartridges

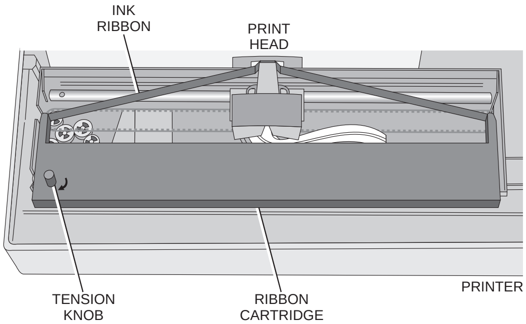
The single item in a dot-matrix printer that requires the most attention is the `ribbon cartridge`. The ink ribbon is wrapped on a spool inside the cartridge and moves across the face of the platen, as depicted in Figure 10.23. A take-up wheel draws new ribbon out of the spool as it is used. As the ribbon wears out, the printing will become faint and uneven. When the print becomes noticeably faint, the cartridge should be replaced. Most dot-matrix printers use a snap-in ribbon cartridge.

To replace a typical ribbon cartridge, move the printhead carriage assembly to the center of the printer. Remove the old cartridge by freeing it from its clips or holders and lifting it out of the printer.

Tighten the ribbon tension by advancing the tension knob on the cartridge in a counterclockwise direction until the ribbon is taut. Snap the cartridge into place, making certain that the ribbon slides between the printhead and the ribbon mask. Slide the printhead assembly back and forth on the rod to check for proper ribbon movement.

Figure 10.23

The printer cartridge.



Paper Specifications

Another reason for faint printing is that the paper-thickness lever is set to the wrong position for the weight of paper being used.

Paper is specified in terms of its `weight` per 500 sheets at 22×17 inches. (That is, 500 sheets of 22×17-inch, 21-pound bond paper weighs 21 pounds.) The thickness setting will also cause smudged characters when the paper is too thick for the actual setting. In this case, adjust the thickness lever one or two notches away from the paper.

Printhead Not Printing

If the printhead is moving but not printing, begin by checking the printer's `Head-Gap lever` to make sure that the printhead is not too far back from the paper. If the printhead does not operate, components involved include the following:

- The printhead
- The flexible signal cable between the control board and the printhead
- The control board
- Possibly the power-supply board

Run the printer's self-test to see whether the printhead will print from the onboard test. Check the flexible signal cable to make sure it is firmly plugged into the control board and that it is not damaged or worn through. If none of the print wires are being energized, the first step should be to exchange the control board for a known good one of the same type. If the new control board does not correct the problem, replace the printhead. A power-supply problem could also cause the printhead to not print.

A related problem occurs when one or more of the print wires does not fire. If this is the case, check the printhead for physical damage. Also check the flexible, signal cable for a broken conductor. If the control board is delivering any of the other print wire signals, the problem is most likely associated with the printhead mechanism. Replace the printhead as a first step. If the problem continues after replacing the printhead, however, exchange the control board for a new one.

To exchange the printhead assembly, make sure that the printhead assembly is cool enough to be handled. These units can get hot enough to cause a serious burn. Unplug the printhead assembly from the control board. Slide the printhead assembly to the center of the printer and rotate the `Head-Locking lever` to release the printhead from the assembly. Remove the printhead by lifting it straight up. Install the new printhead by following the disassembly procedure in reverse. Adjust the new printhead for proper printing. If the tops of characters are missing, the printhead is misaligned with the platen. It may need to be reseated in the printhead carriage, or the carriage assembly may need to be adjusted to the proper height and angle.

It may become necessary to adjust the printhead mechanism to obtain proper printing. This procedure is illustrated in Figure 10.24. To print correctly, the printhead should be approximately 0.6 mm from the platen when the head position lever is in the center position. Move the printhead to the center of the printer. Adjusting this setting requires that the nut at the left end of the rear carriage shaft be loosened. Using a feeler gauge, set the distance between the platen and printhead (not the ribbon mask). Tighten the nut and check the spacing between the printhead and platen at both ends of the printhead travel.

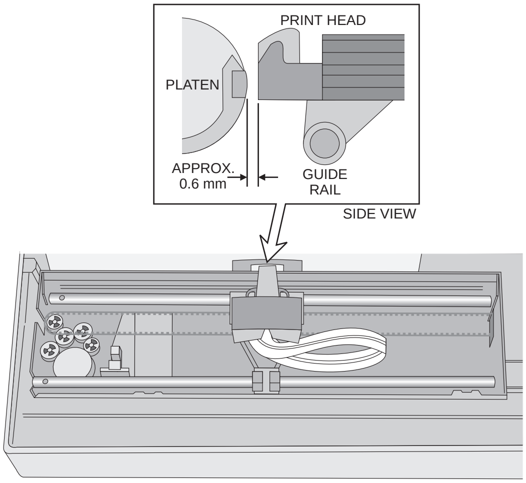


Figure 10.24

Adjusting the printhead spacing.

Finally, check the distance between the platen and the ribbon mask. This spacing should be 0.3 mm. If not, loosen the screws that hold the ribbon mask to the printhead assembly and adjust the gap with feeler gauges. There should also be a 0.1 mm spacing between the printhead and the ribbon mask. After setting the various gaps, run the printer's self-test to check for print quality.

Printhead Not Moving

If the printhead is printing but not moving across the page, a single block of print will be generated on the page. When this type of problem occurs, the related components include the printhead-positioning motor, the timing belt, the `home position and timing sensors`, the control board, and possibly the power-supply board.

With the power off, manually move the printhead to the center of the printer. Turn the printer on to see whether the printhead seeks the home position at the far-left side of the printer. If it moves to the left side of the printer and does not shut off or does not return to the center of the printer, the home position sensor is malfunctioning and should be replaced. If the printhead moves on startup, but will not move during normal printing, the control board should be replaced. In the event that the printhead assembly will not move at any time, the printhead-positioning motor should be replaced. If the print is skewed from left to right as it moves down the page, the printer's bidirectional mode settings may be faulty, or the home position/end-of-line sensors may be defective.

Testing the timing sensor requires test equipment, in the form of a logic probe or an oscilloscope, to look for pulses produced as the printhead is manually moved across the printer.

Figure 10.25 depicts the components associated with the printhead's timing belt. Replacing the timing belt requires that it be removed from the printhead assembly. In many cases, the belt is secured to the printhead assembly with adhesive cement. This requires that the belt's adhesive seal be cut with a single-edged razor blade or a hobby knife. After the seal has been broken, it should be possible to shove the belt out of the clips that secure it to the printhead assembly. Next, remove the belt from the drive-pulley assembly at the positioning motor. It may be necessary to remove the positioning motor from the case to gain access to the pulley.

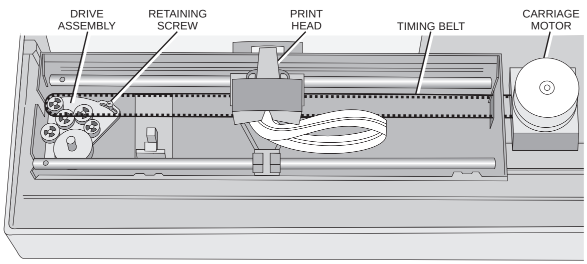


Figure 10.25

Printhead timing.

To reinstall the timing belt, apply a small drop of adhesive to the belt and reattach it to the printhead assembly. Wrap the belt around the positioning motor's drive pulley and reinstall the motor. Following this, it will be necessary to adjust the tension on the belt. To set the tension on the belt, loosen the adjustment screw on the belt-tension adjustment plate. Tighten the timing belt until it will not move more than 1/4 inch when the printhead is at either end of the carriage shaft and the belt is pressed inward. Tighten the retaining screw to lock the tension plate in place. Run the printer's self-test and check the distance between the characters. If the intercharacter spacing is not uniform, replace the belt and perform the check again.

Paper Not Advancing

When the paper does not advance, the output will normally be one line of dark blocks across the page. Examine the printer's `Paper Feed Selector lever` to make sure that it is set properly for the type of paper feed selected (for example, friction feed, pin feed, or tractor feed). If the paper feed is set correctly, the printer is on-line, and the paper will not move, it will be necessary to troubleshoot the paper handling motor and gear train. Check the motor and gear train by setting the printer to the offline mode and holding down the Form Feed (FF) button.

If the feed motor and gear train work from this point, the problem must exist on the control board, with the `interface cable`, the printer's `configuration`, or the `computer system`. If the motor or gear train does not respond, unplug the paper-feed motor cable and check the resistance of the motor windings. If the windings are open, replace the paper-feed motor.

To replace the paper-feed motor or gear train, remove the screws that hold the paper-feed motor to the frame of the printer. Create a wiring diagram that describes the routing of the feed motor's wiring harness. Disconnect the wiring harness from the control board.

Prepare a drawing that outlines the arrangement of the gear train (if multiple gears are used). Remove the gear(s) from the shaft(s), taking care not to lose any washers or springs that may be located behind the gears. After reinstalling the gears and new motor, adjust the motor and gear relationships to minimize the gear lash so that they do not bind or lock up. Use the printer's self-test to check the operation of the motor and gears. Never lubricate the gear train or platen assembly of a dot-matrix printer.

Ink-Jet Printers

Objectives

Ink-jet printers produce characters by squirting a precisely controlled stream of ink drops onto the paper, as described in Figure 10.26. The drops must be controlled very precisely in terms of their aerodynamics, size, and shape; otherwise, the drop placement on the page becomes inexact, and the print quality falters.

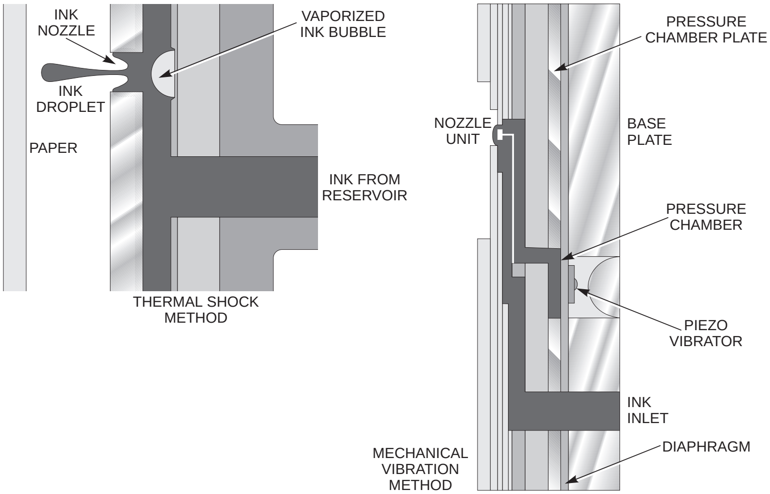


Figure 10.26

Ink-jet printers.

The drops are formed by one of two methods:

- Thermal shock—Heats the ink in a capillary tube, just behind the nozzle. This increases the pressure of the ink in the tube and causes it to explode through the opening.
- Mechanical vibration—Uses vibrations from a piezo-electric crystal to force ink through a nozzle.

Ink-jet printers use two methods to deliver the drops to the page: the `interrupted-stream` (drop-on-demand) method, and the `continuous-stream` method. The drop-on-demand method forms characters on the page in much the same manner as a dot-matrix printer does. As the printhead mechanism moves across the character cells of the page, the controller causes a drop to be sprayed—only where necessary—to form the dot pattern of a character. Drop-on-demand printing is illustrated in Figure 10.27.

Continuous-stream systems, like the one described in Figure 10.28, produce characters that more closely resemble fully formed characters. In these systems, the printhead does not travel across the page. Instead, the drops are given a negative charge in an ion chamber and passed through a set of deflection plates, similar to the electron beam in a CRT tube. The plates deflect the drops to their proper placement on the page, and unused drops are deflected off the page into an ink recirculation system.

Figure 10.27

Drop-on-demand printing.

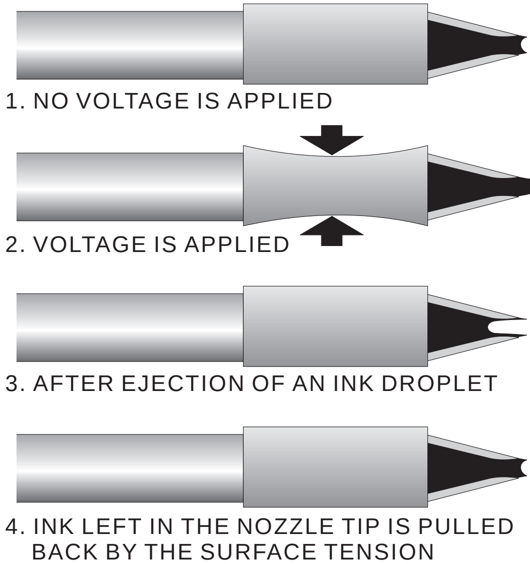
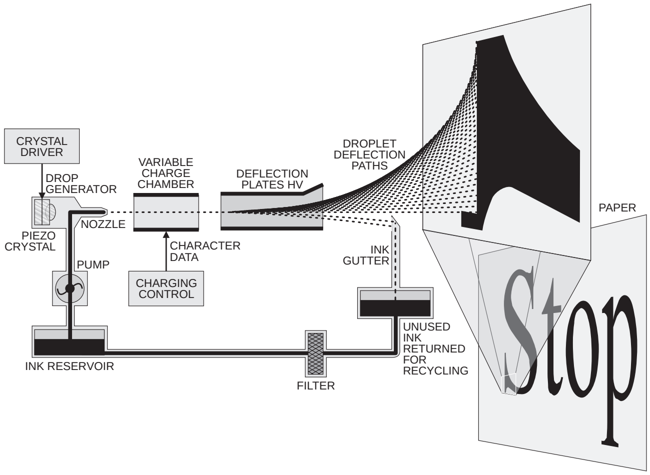


Figure 10.28

Continuous-stream printing.



Although capable of delivering very high-quality characters at high speeds, continuous-stream systems tend to be expensive and therefore are not normally found in printers for the consumer market. Instead, they are reserved for high-volume, commercial applications. The ink-jet printers in this market all use drop-on-demand techniques to deliver ink to the page.

Some ink-jet printers incorporate multiple jets to permit color printing. Four basic colors may be mixed to create a veritable palate of colors by firing the ink jets in different combinations.

Ink-Jet Printer Components

Aside from the printing mechanism, the components of a typical ink-jet printer are very similar to those of a dot-matrix printer. Its primary components are as follows:

- The printhead assembly
- The power board
- The control board
- The printhead-positioning motor and timing belt
- The paper-feed motor and gear train
- The printer's sensors

These components are described in Figure 10.29.

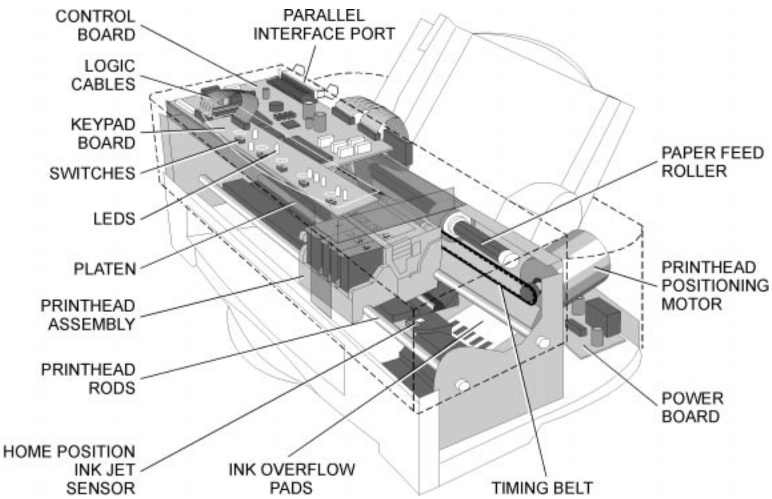


Figure 10.29

Ink-jet printer components.

The Printhead Assembly

The ink cartridge snaps into the printhead assembly that rides in front of the platen on a rail or rod. The printhead assembly is positioned by a timing belt that runs between it and the positioning motor. A flexible cable carries ink-jet firing information between the control board and the printhead. This cable folds out of the way as the printhead assembly moves across the printer.

Paper Handling

The paper-feed motor turns a gear train that ultimately drives the platen, as depicted in Figure 10.30. The paper is friction-fed through the printer, between the platen and the pressure rollers. Almost all ink-jet printers used with microcomputer systems are single-sheet, friction-feed systems. The control board, power-supply board, and sensors perform the same functions in an ink-jet printer that they do in the dot-matrix printer.

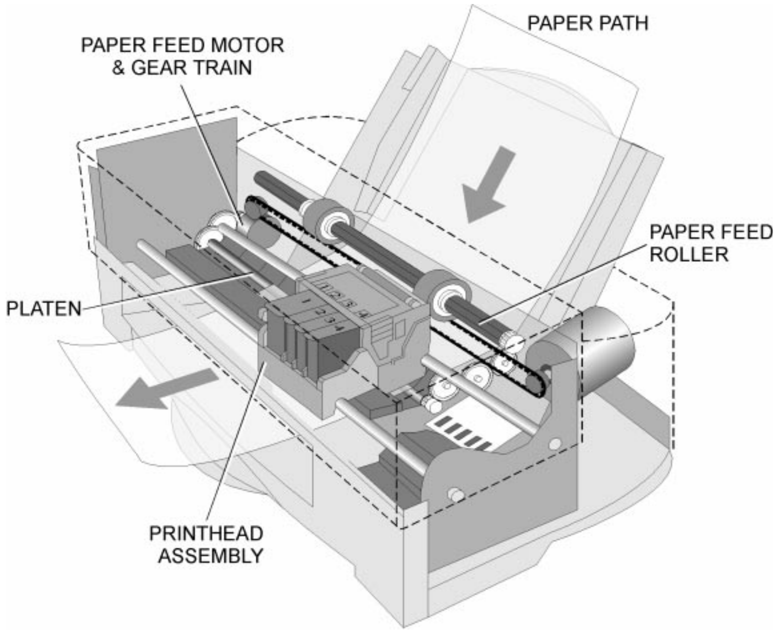


Figure 10.30

Ink-jet paper handling.

Troubleshooting Ink-Jet Printers

Objectives

As with the dot-matrix printer, the first step in determining the cause of an ink-jet printer problem is to determine which part of the printer system is at fault—the computer, the cable, or the printer. Ink-jet printers are equipped with built-in self-tests. The easiest way to determine whether the printer is at fault is to run its self-tests. Consult the printer's User's Manual for instructions on running its self-tests.

If the printer runs the self-tests and prints clean pages, most of the printer has been eliminated as a possible cause of problems. The problem could be in the computer, the cabling, or the interface portion of the printer. If the printer fails the self-tests, however, it will be necessary to diagnose the printer problem. The following section presents typical problems encountered in ink-jet printers.

The following are symptoms of ink-jet printer problems:

- No lights or noise from printer.
- Light or uneven print being produced.
- Printhead moving, but not printing, or printing erratically.
- Lines on the page.
- Printhead printing, but does not move.
- Paper will not advance.

Ink-Jet Printer Configuration Checks

The presence of the printer's onboard microcontroller allows modern printers to be very flexible. Like other peripheral devices, printers can be configured to operate in different modes. Operating configuration information can be stored in RAM, on the control board.

In the case of ink-jet printers, the configuration settings are normally entered into the printer through software. Typical configuration information includes the following:

- Page orientation (landscape or portrait)
- Paper size
- Collation
- Print quality

Landscape printing is specified when the width of the page is greater than the length of the page. Portrait printing is specified when the length of the page is greater than the width. In an ink-jet printer, the quality of the print is specified in the number of dots per inch produced. Typical ink-jet resolutions run from 180×180 dpi to 720×720 dpi. Ink-jet printers have the capability to download additional fonts from the computer.

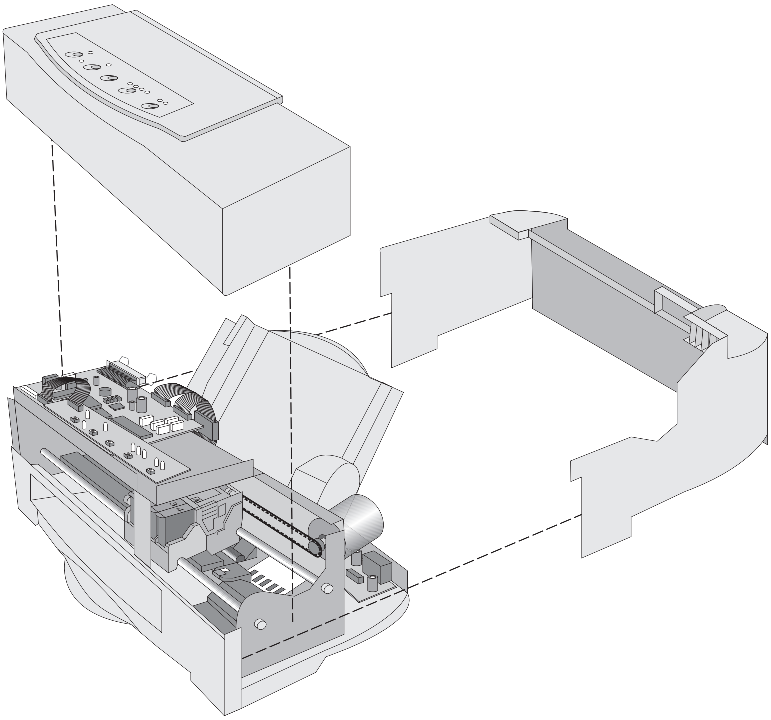
You can also configure the basic appearance of color and `grayscale` images produced by the ink-jet printer. A color ink-jet printer uses four ink colors to produce color images. These are cyan, magenta, yellow, and black (referred to as CMYK color). To create other colors, the printer prints a predetermined percentage of the basic colors in close proximity to one another. The different percentages determine what the new color will be. The eye does not differentiate the space between them, and perceives only the combined color. This is referred to as `halftone color`. Typical color configurations include setting up the brightness, the contrast, and the saturation settings of images.

Ink-Jet Printer Hardware Checks

To perform work on the printer's hardware, it is necessary to disassemble the printer's case. Begin by removing all the add-on pieces such as dust covers and paper feeders. Remove the screws that hold the outer panels of the case to the printer frame. Removing the access panels of a typical ink-jet printer is described in Figure 10.31. The retaining screws are sometimes hidden beneath rubber feet and compliance stickers. Finally, it may be necessary to disconnect the printer's front-panel connections from the control board to complete the disassembly of the case.

Figure 10.31

Printer Case.



Power-Supply Problems

If the printer will not function and displays no lights, no sounds, and no actions, the power supply is generally involved. Check the Online light. If the printer is offline, no print action will occur. A missing or improperly installed ink cartridge will prevent the unit from printing. Install the ink cartridge correctly. Check the power outlet to make certain that it is live. Plug a lamp or other device in the outlet to verify that it is operative. Check to see that the power cord is plugged in securely to the printer and the socket. Make sure the power switch is turned on.

If the unit is plugged in and turned on, but still not working, turn it off and unplug it. Remove the top of the printer's case and locate the power-supply board. Check the power-supply's fuse to make sure that it is good. If the fuse is blown, replace it with a fuse of the same type and rating. Do not replace a blown fuse with a conductor or a slow-blow fuse. To do so could lead to more extensive damage to the printer, and possible unsafe conditions.

Also, check the power supply and control boards, as well as the paper-feed and printhead-positioning motors, for burnt components or signs of defect. Fuses do not usually blow unless another component fails. The other possible cause of overcurrent occurs when a motor (or its gear train) binds and cannot move. Check the drive mechanisms and motors for signs of binding. If the gear train or positioning mechanisms will not move, they may need to be adjusted or replaced before replacing the fuse.

If none of the printer sections work, everything is connected, and power is applied, it will be necessary to exchange the power-supply board for a new unit. Unlike the computer's power supply, the typical power supply in a printer is not enclosed in a protective housing and, therefore, presents a shock hazard any time it is exposed.

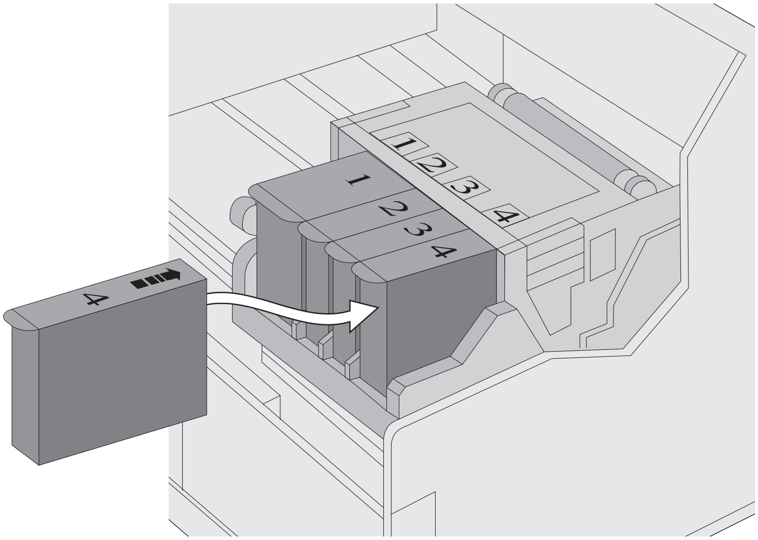
To exchange the power-supply board, disconnect the power cable from the printer. Disconnect and mark the cabling from the control board and any other components directly connected to the power supply. Remove any screws or clips that secure the power-supply board to the case. Lift the board out of the cabinet. Install the new board and reconnect the various wire bundles to it.

Ink Cartridges

The single item in an ink-jet printer that requires the most attention is the `ink cartridge` (or cartridges). As the ink cartridge empties, the printing will eventually become faint and uneven, and the resolution of the print on the page will diminish. When the print becomes noticeably faint, or the resolution becomes unacceptable, the cartridge will need to be replaced. Most ink-jet printers use a self-contained, snap-in ink cartridge, like the one in Figure 10.32. Some models have combined ink cartridges that replace all three colors and the black ink at the same time. Other models use individual cartridges for each color. In this way, only the colors running low are replaced.

Figure 10.32

Self-contained, snap-in ink cartridge.



The ink cartridges can be popped out of the printhead assembly to inspect its ink jets. If any, or all, of the jets is clogged, it is normally possible to clear them by gently wiping the face of the cartridge with a swab. A gentle squeeze of the ink reservoir can also help to unblock a clogged jet. Using solvents to clear blockages in the jets can dilute the ink and allow it to flow uncontrollably through the jet.

To replace a typical ink cartridge, move the printhead carriage assembly to the center of the printer. Remove the old cartridge by freeing it from its clips or holders and lifting it out of the printer.

Printhead Not Printing

If the printhead is moving but not printing, begin by checking the ink supply in the print cartridge. The reservoir does not have to be completely empty to fail. Replace the cartridge(s) that appears to be low. Some or all of the jets may be clogged. This is particularly common if the printer has not been used for a while. If there are cleaning instructions in the User's Manual, clean the jets and attempt to print from the self-test.

If the printer will not print from the self-tests, the components involved include the following:

- The printhead
- The flexible signal cable (between the control board and the printhead)
- The control board
- Possibly the power-supply board

Check the flexible signal cable to make sure that it is firmly plugged into the control board and that it is not damaged or worn through. If none of the ink jets are firing, the first step should be to exchange the ink cartridges for new ones. If a single ink jet is not firing, replace the cartridge that is not working.

Next, use the ohmmeter function of a multimeter to check the continuity of the conductors in the flexible-wiring harness that supplies the printhead assembly. If one of the conductors is broken, a single jet will normally be disabled. If the broken conductor is a ground or common connection, however, all the jets should be disabled. Exchange the control board for a known good one of the same type. If the new control board does not correct the problem, replace the printhead. A power-supply problem could also cause the printhead to not print.

If a single jet is not functioning, the output will appear as a white line on the page. If one of the jets is activated all the time, black or colored lines will be produced on the page. Use the previous steps to isolate the cause of these problems—replace the print cartridge, check the flexible cabling for continuity and for short circuits between adjacent conductors, exchange the control board for a known good one, and finally, check the power supply.

Printhead Not Moving

If the printhead is printing but not moving across the page, a single block of print will normally be generated on the page. When this type of problem occurs, the related components include the printhead-positioning motor, the timing belt, the home position sensor, the control board, and possibly the power supply. These components are depicted in Figure 10.33.

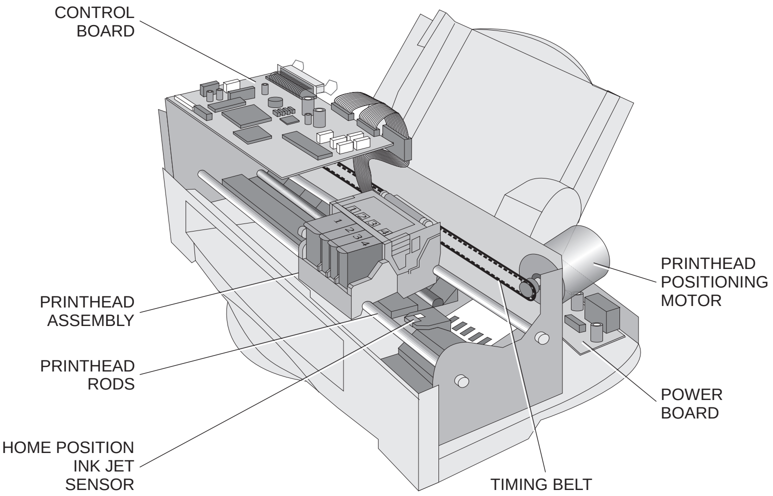


Figure 10.33

Printhead Positioning Components.

With the power off, manually move the printhead to the center of the printer. Turn the printer on to see whether the printhead seeks the home position at the far end of the printer. If the printhead moves to the end of the printer and does not shut off or does not return to the center of the printer, the home position sensor is malfunctioning and should be replaced. If the printhead moves on startup, but will not move during normal printing, the control board should be replaced. In the event that the printhead assembly will not move at any time, check to see if the printer is in Maintenance Mode. In this mode, the printer typically keeps the printhead assembly in the home position. If no mode problems are present, the printhead-positioning motor should be replaced.

If characters are unevenly spaced across the page, the timing sensor may be failing. To test the timing sensor requires test equipment, in the form of a logic probe or an oscilloscope, to look for pulses produced as the printhead is manually moved across the printer.

Replacing the timing belt requires that the belt be removed from the printhead assembly. In many cases, the belt will be secured to the printhead assembly with adhesive cement. This requires that the belt's adhesive seal be cut with a single-edged razor blade or a hobby knife. After the seal has been broken, it should be possible to shove the belt out of the clips that secure it to the printhead assembly. Next, remove the belt from the drive-pulley assembly at the positioning motor. It may be necessary to remove the positioning motor from the case to gain access to the pulley.

Paper Not Advancing

When the paper does not advance, the output will normally be a thick, dark line across the page. Check the Control Panel to see that the printer is online. If the printer is online, and the paper will not move, it will be necessary to troubleshoot the paper handling motor and gear train. Check the motor and gear train by setting the printer to the offline mode and holding down the Form Feed (FF) button.

If the feed motor and gear train work from this point, the problem must exist on the control board, the interface cable, the printer configuration, or the computer system. If the motor or gear train does not respond, unplug the paper-feed motor cable and check the resistance of the motor windings. If the windings are open, replace the paper-feed motor.

To replace the paper-feed motor or gear train, remove the screws that hold the paper-feed motor to the frame of the printer. Create a wiring diagram that describes the routing of the feed motor's wiring harness. Disconnect the wiring harness from the control board.

Draw an outline of the gear train arrangement (if multiple gears are used). Remove the gear(s) from their shafts, taking care not to lose any washers or springs that may be located behind the gears. After reinstalling the gears and new motor, adjust the motor and gear relationships to minimize the gear lash so that they do not bind or lock up. Use the printer's self-tests to check the operation of the motor and gears.

Laser Printers

Objectives

The laser printer modulates a highly focused laser beam to produce CRT-like raster-scan images on a rotating drum, as depicted in Figure 10.34. The drum is coated with a photosensitive plastic, which is given a negative electrical charge over its surface. The modulated laser beam creates spots on the rotating drum. The spots written by the laser take on a positive electrical charge. A negatively charged toner material is attracted to the positively charged, written areas of the drum. The paper is fed past the rotating drum and the toner is transferred to the paper. A pair of compression rollers and a high temperature lamp fuse the toner to the paper. Thus, the image, written on the drum by the laser, is transferred to the paper.

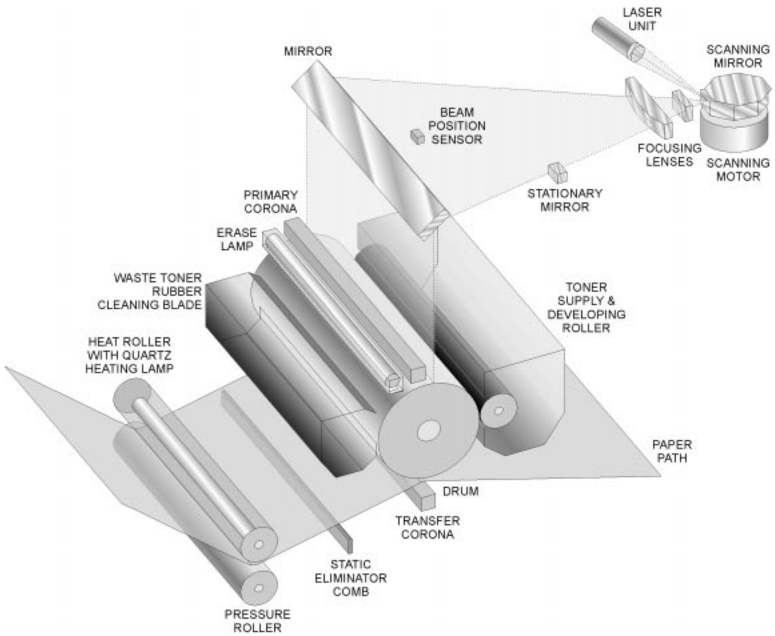


Figure 10.34

A typical laser printer.

The laser beam scans the drum so rapidly that it is not practical to do the scanning mechanically. Instead, the beam is bounced off a rotating, polygonal (many-sided) mirror. The faces of the mirror cause the reflected beam to scan across the face of the drum as the mirror revolves. Using the highest dot densities available, these printers produce characters which rival type-set text. Larger laser printers produce characters at a rate of 20,000 lines per minute. Laser printers intended for the personal computer market generate 6 to 45 pages per minute.

Laser-Printer Components

From manufacturer-to-manufacturer, and model-to-model, the exact arrangement and combinations of components may vary in laser printers. However, the order of operations is always the same. The six stages of operation in a laser printer include the following:

- Cleaning
- Conditioning
- Writing
- Developing
- Transferring
- Fusing

To accomplish these objectives, all laser printers possess the following logical blocks:

- Power supply
- Control board
- Laser writing unit
- Drum unit
- Fusing assembly
- Paper-feed motor and gear train
- System's sensors
- Control Panel board

The blocks of the typical laser printer are illustrated in Figure 10.35.

The laser printer power-supply unit is the most complex found in any type of printer. It must deliver AC power to the fuser unit. This unit requires power for its fusing heaters and image-erase lamps. The power supply also delivers a high-voltage DC supply (+1000 Vdc) to the toner-transfer mechanisms in the drum area.

The high voltages are used to create the static charges required to move toner from one component to another (that is, from the drum to the paper). Finally, the power-supply unit must deliver DC operating voltages to the scanning and paper-handling motors and the digital electronic circuitry on the control board.

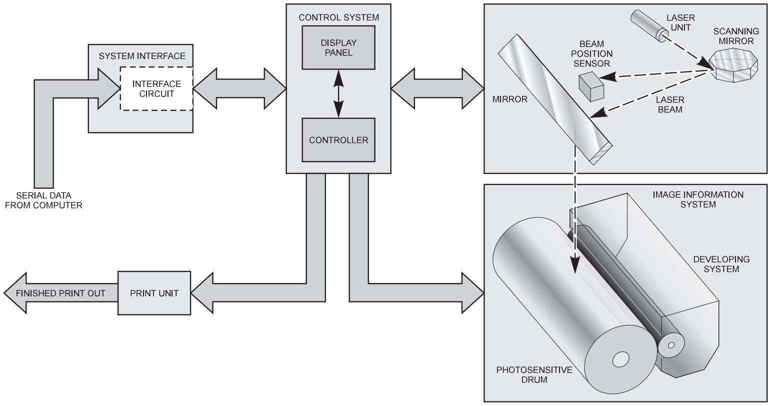


Figure 10.35

Block diagram of a laser printer.

The control board contains all the circuitry required to operate the printer and control its many parts. It receives control signals from the computer and formats the data to be printed. The control board also monitors the conditions within the printer and responds to input from the its various sensors.

When data is received from the host computer, the control board generates all the enabling signals to place the information on the page as directed. The character information is converted into a serial bit stream, which can be applied to the scanning laser. The photosensitive drum rotates as the laser beam is scanned across it. The laser creates a copy of the image on the photosensitive drum in the form of a relatively positive-charged drawing. This operation is referred to as `registration`.

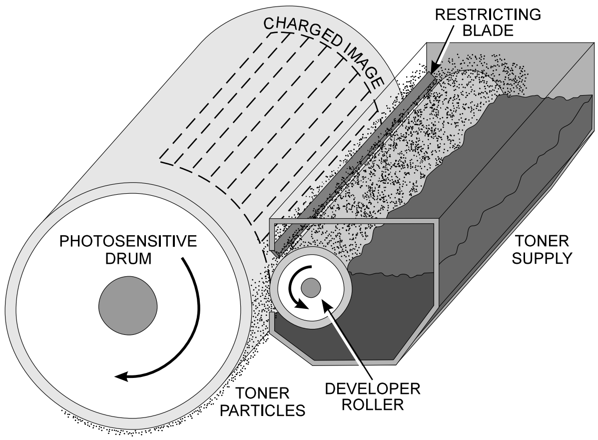
Laser-Printing Operations

Before the laser writes on the drum, a set of erase lamps shine on the drum to remove any residual traces of the preceding image. This leaves the complete drum with a neutral electrical charge. A high voltage applied to the primary corona wire creates a highly charged negative field that `conditions` the drum to be written on, by applying a uniform negative charge (–600 V) to it.

As the drum is written on by the laser, it turns through the `toner powder`, which is attracted to the charged image on the drum. Toner is a very fine powder bonded to iron particles that are attracted to the charges written on the drum. The developer roller in the toner cartridge turns as the drum turns and expels a measured amount of toner past a restricting blade, as illustrated in Figure 10.36. A regulating AC voltage assists the toner in leaving the cartridge, but also pulls back some excess toner from the drum. Excess toner is recycled within the toner cartridge so that it can be used again.

Figure 10.36

The developer roller.



Great care should be taken when installing a new drum unit. Exposing the drum to light for more than a few minutes may damage it. The drum should never be touched; this, too, can ruin its surface. Keep the unit away from dust and dirt as well as away from humidity and high-temperature areas.

The `transfer corona wire` (transfer roller) is responsible for transferring the toner from the drum to the paper. The toner is transferred to the paper because of the highly positive charge the transfer corona wire applies to the paper. The positive charge attracts the negative toner particles away from the drum and onto the page. A special, static-eliminator comb acts to prevent the positively charged paper from sticking to the negatively charged drum.

After the image has been transferred to the paper, a pair of `compression rollers` in the fusing unit act to press the toner particles into the paper, while they melt them to it. The top compression roller, known as the `fusing roller`, is heated by a quartz lamp. This roller melts the toner to the paper as it exits the unit, while the lower roller applies pressure to the paper. A cleaning pad removes excess particles and applies a silicon lubricant to the roller to prevent toner from sticking to the Teflon-coated fusing roller. A demonstration of the complete transfer process is given in Figure 10.37.

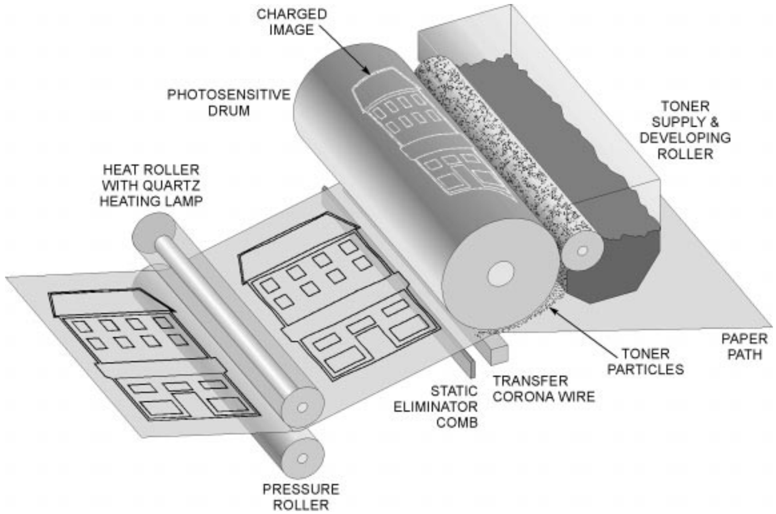


Figure 10.37

The transfer process.

Component Variations

In Hewlett-Packard printers, the main portion of the printing system is contained in the electrophotographic cartridge. This cartridge contains the toner supply, the corona wire, the drum assembly, and the developing roller. The HP configuration is depicted in Figure 10.38.

In other laser printers, such as the one depicted in Figure 10.39, the basic components are combined so that the printer consists of a developer unit, a toner cartridge, a drum unit, a fuser unit, and a cleaning pad. In this case, the developer unit and toner cartridge are separate units. With this configuration, changing the toner does not involve changing some of the other wear-prone components. Although it is less expensive to change toner, attention must be paid to how much the other units are wearing. Notice that the photosensitive drum is also a separate component.

Figure 10.38

The HP cartridge configuration.

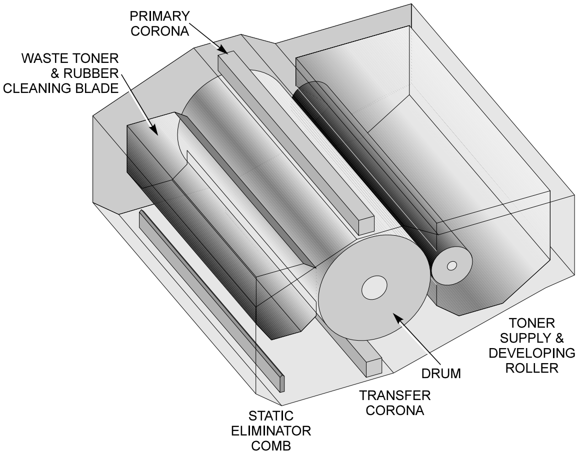
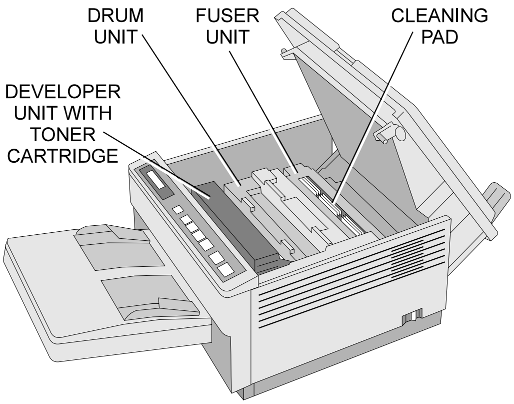


Figure 10.39

Basic components of a laser printer.



Paper Handling

Laser printers are very mechanical in nature. The paper-handling motor and the gear train assembly perform a tremendous number of operations to process a single sheet of paper. The paper-transport mechanics must pick up a page from the paper tray and move it into the printer's registration area. After the drum has been written with the image, the paper-handling mechanism moves the paper into registration. A roller system moves the page past the drum and into the fusing unit. When the page exits through the fusing rollers, the printer senses that the page has exited and resets itself to wait for another page to print.

In addition to the motor and gear train, the printer uses a number of sensors and solenoid-actuated clutches to control the paper movement. It uses solenoids to engage and disengage different gear sets and clutches at appropriate times during the printing process.

A typical laser printer will have sensors to determine what paper trays are installed, what size paper is in them, and whether the tray is empty. It will also use sensors to track the movement of the paper through each stage of the printer. This allows the controller to know where the page is at all times, and to sequence the activities of the solenoids and clutches properly. Figure 10.40 gives a summary of the sensors found in a typical laser printer.

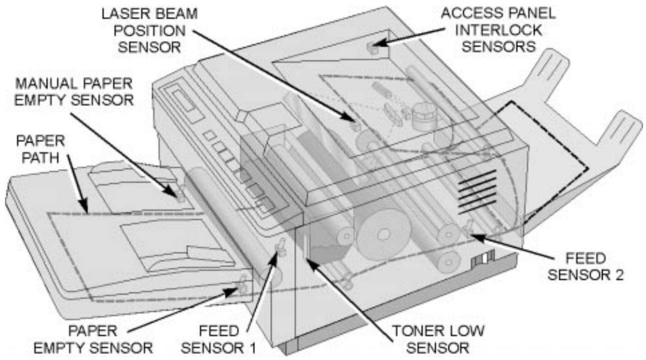


Figure 10.40

Sensor summary.

If the page does not show up at the next sensor at the appropriate time, the printer will know that a paper jam has occurred and will create an error message that indicates the area of the printer where it is. When a paper jam occurs, it will be necessary to remove the paper from the inside of the printer and reset the print operation. Gaining access to the area of the printer where the jam is usually requires direction from the printer's User's manual. The printer should always be allowed to cool, and should always be turned off before reaching inside the unit.

Another set of sensor switches monitor the printer's access doors to protect personnel from potentially dangerous conditions inside the printer. The `Interlock switch` blocks the laser beam as a vision-protection measure. Likewise, the high-voltage supplies to various printer components are also shut down. To observe the operation of the printer, it is necessary to locate and bypass these interlocks. You should always be aware that these interlocks are present for protection, however, and great care should be taken when working with them defeated.

Still other sensors are used to monitor the temperatures within different sections of the printer. A `thermal sensor` in the fusing unit monitors the temperature of the unit. This information is applied to the control circuitry, so that it can control the fuser temperature between 140°C and 230°C. If the temperature of the fuser is not controlled correctly, it may cause severe damage to the printer as well as present a potential fire hazard. A thermal fuse protects the fuser assembly from overheating and damaging the printer. The thermal fuse should normally snap back after the temperature condition is cleared. If the switch is open under cool conditions, it will need to be replaced. This is normally an indication that the thermal sensor has failed, or that the fuser assembly has been installed improperly.

When the laser beam is turned on, a `beam detector sensor` in the writing unit alerts the control circuitry that the writing process has begun. This signal synchronizes the beginning of the laser-modulating data with the beginning of the scan line.

Troubleshooting Laser Printers

Objectives

Many of the problems encountered in laser printers are similar to those found in other printer types. Notice, for example, that most of the symptoms listed in the following section relate to the printer not printing, or not printing correctly, and not moving paper through the printer.

Because of the extreme complexity of the laser printer's paper-handling system, paper jams are a common problem. This problem tends to increase in frequency as the printer's components wear from use. Basically, paper jams occur in all three main sections of the printer. These areas are as follows:

- The pickup area
- The registration area
- The fusing area

If the rubber separation pad in the pickup area is worn excessively, more than one sheet of paper may be drawn into the printer, causing it to jam. Also, if additional paper handling features, such as duplexers (for double-sided copying) and collators (for sorting) are added, they will contribute to the possibility of jams as they wear. Paper problems can also cause jams to occur. Using paper that is too heavy, or too thick, can result in jams, as can overloading paper trays. Similarly, using the wrong type of paper can defeat the separation pad and allow multiple pages to be drawn into the printer, resulting in a jam. Using coated paper stock can be hazardous because the coating might melt or catch fire.

Unlike other printer types, the laser printer tends to have several high-voltage and high-temperature hazards inside it. To get the laser printer into a position where you can observe its operation, it is necessary to defeat some interlock sensors. This action will place you in potential contact with the high-voltage, high-temperature areas previously mentioned. Take great care when working inside the laser printer.

The following are symptoms of laser printer problems:

- Printer dead, power on, but no printing.
- The print on the page is light or washed out.
- A blank page is produced.
- Stains or black dust on paper.
- Vertical lines on paper.
- The printer will not load paper.
- Paper jams in printer.
- A paper jam has been cleared, and the unit still indicates a jam is present.

Laser-Printer Configuration Checks

Like other complex peripheral equipment, laser printers must be configured for the desired operational characteristics. The printer is an extension of the computer system and, therefore, must be part of the overall configuration. To make the system function as a unit, configure the computer, configure the printer, and configure the software. Review and record the computer's configuration information for use in setting up the printer and software. Configure the printer with the parameters that you want it to use, record these settings, and then set up the software to match. Consult the printer's User's Manual for configuration information specific to setting up that particular printer.

Laser-Printer Hardware Checks

Variations in the hardware organization of different laser printers makes it impossible to write a general troubleshooting routine that can be applied to all them without being specific to one model. The following troubleshooting discussions are general, and will require that the user do some interpretation to apply them to a specific laser printer.

Fortunately, laser-printer hardware has become highly modularized, as described in Figures 10.39 and 10.40. This allows entire sections of hardware to be checked by changing a single module. Unfortunately, the mechanical gear train and sensor systems are not usually parts included in the modules. Therefore, their operation will need to be checked individually.

Printer Is Dead or Partially Disabled

As usual, when the printer appears to be dead, the power supply is suspected. Again, as usual, the power supply can affect the operation of basically every section of the printer. In the laser printer, this is particularly complicated because three types of power are being delivered to the various printer components.

If the printer does not start up, check all the normal, power-supply–related check points (that is, power cord, power outlet, internal fuses, and so on). If the printer's fans and lights are working, other components that are associated with a defective power supply include the following:

- Main motor and gear train
- High-voltage corona wires

- Drum assembly
- Fusing rollers

There are four basic reasons why the main motor may not run when the printer is supposed to print. These include the following:

- The portion of the power supply that supplies the motor is defective.
- The control circuitry is not sending the enabling signals to turn the motor on.
- The motor is dead.
- The gear train is bound up, and will not let the motor turn.

In the last case, there should be sounds from the fan running, and lights on the Control Panel. Isolate the failure and troubleshoot the components involved in that section.

If the high-voltage portion of the power supply that serves the corona wires and drum sections is defective, the image delivered to the page will be affected. If the high-voltage section of the power supply fails, the transfers of toner to the drum and then to the paper cannot occur. The contrast control will not be operational either. In cases of partial failure, the image produced will have a washed-out appearance. Replace the high-voltage section of the power supply or the drum unit. If a separate corona wire is used, let the printer cool off sufficiently and replace the wire. Never reach into the high-voltage, high-temperature corona area while power is applied to the printer. Also, avoid placing conductive instruments in this area.

If the DC portion of the power supply fails, the laser beam will not be produced, creating a `Missing Beam` error message. The components involved in this error are the laser/scanning module, the control board, and the DC portion of the power supply. Replace the L/S module, the DC portion of the power supply, and the main control board.

When the heating element or lamp in the fusing area does not receive adequate AC power from the power supply, the toner will not affix to the page as it should. This condition will result in smudged output.

If the printer remains in a constant state of starting up, this is equivalent to the computer not passing the POST tests portion of the bootup process. If the printer starts up to an offline condition, there is likely a problem between the printer and the host computer's interface. Disconnect the interface cable and check to see whether the printer starts up to a ready state. If so, the problem is in the computer, its interface, its configuration, or the cable. Troubleshoot the system in this direction. If the printer still does not start up, note the error message produced and check the sections of the printer related to that section. Check to see whether the printer is connected to the system through a print-sharing device. If so, connect the printer directly to the system, and then try it. It is not a good practice to use laser printers with these types of devices.

A better arrangement is to install, or just use, an LPT2 port to attach an additional printer to the system. Beyond two printers, it would be better to network the printers to the system.

Print on Page Is Missing or Bad

Many of the problems encountered in laser printers are associated with missing or defective print on the page. Normal print delivery problems fall into eight categories, as follows:

- Black pages
- White (blank) pages
- Faint print
- Random specks on the page
- Faulty print at regular intervals on the page
- White lines along the page
- Print missing from some portion of the page
- Smudged print

A black page indicates that toner has been attracted to the entire page. This condition could be caused by a failure of the primary corona, the laser scanning module, or the main control board. If the laser is in a continuous On condition, the entire drum will attract toner. Likewise, if the primary corona is defective, the uniform negative charge will not be developed on the drum to repel toner. Replace the primary corona or drum assembly. If the problem continues, replace the laser scanning module and the main control board.

On the other end of the spectrum, a white page indicates that no information is being written on the drum. This condition basically involves the laser-scanning module, the control board, and the power supply. Another white page fault occurs when the corona wire becomes contaminated or corroded so that the attracting charge between the drum and paper is severely reduced.

Specks and stains on the page may be caused by a worn-out cleaning pad or a defective corona wire. If the cleaning pad is worn, it will not remove excess toner from the page during the fusing process. If the corona wire's grid does not regulate the charge level on the drum, dark spots will appear in the print. To correct these situations, replace the corona assembly by exchanging the toner cartridge or drum unit. Also, replace the cleaning pad in the fusing unit. If the page still contains specks after changing the cartridge, run several pages through the printer to clear excess toner that may have collected in the printer.

White lines along the length of the page are generally caused by poorly distributed toner. Try removing the toner cartridge and gently shaking it to redistribute the toner in the cartridge. Other causes of white lines include damaged or weakened corona wires. Check and clean the corona wires, if accessible, or replace the module containing the corona wires.

Faint print in a laser printer can be caused by a number of different things. If the contrast control is set too low, or the toner level in the cartridge is low, empty, or poorly distributed, print quality can appear washed-out. Correcting these symptoms is fairly easy; adjust the contrast control, remove the toner cartridge, inspect it, shake it gently (if it is a sealed unit), and retry it. If the print does not improve, try replacing the toner cartridge. Other causes of faint print include a weakened corona wire or a weakened high-voltage power supply that drives it. Replace the unit that contains the corona wire. Replace the high-voltage power supply. Make sure that latent voltages have been drained off the high-voltage power supply before working with it.

Faults in the print that occur at regular intervals along the page are normally caused by mechanical problems. When roller and transport mechanisms begin to wear in the printer, bad registration and print appear in cyclic form. This can be attributed to the dimensions of cyclic components such as the drum, developing roller in the toner cartridge, or fusing rollers. Examine the various mechanical components for wear or defects.

Missing print is normally attributed to a bad or misaligned laser-scanning module. If this module is not correctly installed, it cannot deliver lines of print to the correct areas of the page. Likewise, if the scanning mirror has a defect or is dirty, portions of the print will not be scanned on the drum. Another cause of missing print involves the toner cartridge and low, or poorly distributed, toner. If the toner does not come out of the cartridge uniformly, areas of missing print can be created. A damaged or worn drum can also be a cause of repeated missing print. If areas of the drum will not hold the charge properly, toner will not transfer to it, or the page, correctly.

Smudged print is normally a sign of a failure in the fusing section. If the fusing roller's temperature or pressure is not sufficient to bond the toner to the page, the print will smudge when touched. Examine the fuser unit, the power supply, and the fusing roller's heating unit.

Paper Will Not Feed, or Is Jammed

If the paper will not feed at all, the place to begin checking is the paper tray area. The paper trays have a complex set of sensors and pickup mechanisms that must all be functioning properly to begin the paper handling. Because of the complexity of the paper pickup operation, jams are most likely to occur in this area. Check the paper tray to make sure that there is paper in it and that it has the correct size of paper in it. Each tray in a laser printer has a set of tabs that contact sensor switches, to tell the control circuitry that the tray is installed and what size paper is in it. A mechanical arm and photo detector are used to sense the presence of paper in the tray.

If the printer's display panel indicates a `Paper Out Error` message, locate and actuate the paper detector by hand (lift it up). While holding the paper sensor, check the sensor switches by pressing each one individually. If the `Paper Out` message does not go out when any of the individual switches is pressed, replace that switch. If none of the switches are good, replace the paper sensor and arm. Also, check the springloaded plate in the bottom of the tray to make sure that it is forcing paper up to the pickup roller when the tray is installed in the printer.

The paper pickup roller must pull the top sheet of paper off of the paper stack in the tray. The controller actuates a solenoid that engages the pickup roller's gear train. The pickup roller moves the paper into position against the registration rollers. If the printer's display panel shows a jam in the pickup area, check to make sure that the paper tray is functional, and then begin troubleshooting the pickup roller and main gear train. If none of the gear train is moving, the main motor and controller board need to be checked. The power-supply board may also be a cause of the problem.

If the paper feeds into the printer, but jams after the process has begun, troubleshoot the particular section of the printer where the jam is occurring—pickup, registration, fusing area, and output devices (collators and duplexers). This information is generally presented by the laser printer's display panel. Figure 10.41 describes the paper path through a typical laser printer.

In each stage, you will need to check the action of the gear train in the area. Also, inspect the various rollers in that stage for wear or damage. If the motor and gear train operate, but no action occurs in the pickup roller or registration rollers, check the solenoid and clutches for these units.

Another cause for jams is the presence of some obstruction in the paper path. Check for pieces of paper that have torn loose and lodged in the printer's paper path. In most laser printers, mechanical components are part of a replaceable module (that is, the drum unit, the developing unit, or the fusing unit). If the motor and all the exposed gears are working, replace these units one at a time.

Many times, a paper-jam error will remain even after the paper has been removed from the laser printer. This is typically caused by an interlock error. Just opening the printer's main access door should clear the error.

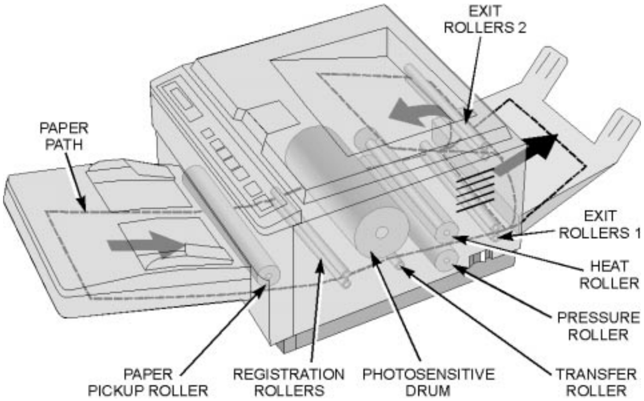


Figure 10.41

The paper path.

Windows-Related Printing Problems

In a Windows-based system, the Windows environment controls the printing function through its drivers. When an application presents a particular font type for printing, Windows must locate or create the font codes. If the code is a TrueType code, it just creates the bitmaps required and sends them to the printer. If the printer code is some other font style, however, Windows must attempt to locate that font in the system.

If the requested font is not available or is not supported by the selected printer, Windows must substitute a font for it. In these cases, the Windows Font Map is used to decide the most appropriate font to use. Windows bases this choice on several factors including the character set, family, typeface, height, and width of the possible substitute font.

When Windows is forced to substitute fonts other than the one called for by the application, printing problems can occur. The printer can lock up or just produce print that is not correct, or that is out of place.

If font-related printing problems are suspected, check to see that TrueType fonts are selected. Some font converters do not work properly with Windows. Therefore, their output is corrupted and will not drive the printer correctly. This should produce a `GPF Fault` message.

The presence of a corrupted font can be determined with the Windows Write text editor. Open the Write program, create a page-long document of MS Sans Serif text, and save the document to disk.

While holding down the left mouse button, drag the mouse across the complete page to select the whole body of text. Click on the Character entry on the toolbar and select the Fonts entry from the drop-down list. Select a font from the list, and print the document. Continue this process, using each font in the list until the defective font is located.

Other factors that can cause font problems include low system RAM and third-party video or printer drivers. A minimum of 1MB RAM is required to print from Windows 3.x, although 2MB is recommended. Use the memory optimization schemes described in Chapter 5 to free up as much memory as possible. Check the video driver setting in the Windows Setup window to determine which video driver is being used. Substitute the standard VGA driver and try to print a document. Check the printer driver using the Control Panel's Print icon to make certain that the correct driver is installed.

Some types of drivers are known to conflict with the Windows TrueType fonts. These include the Adobe Type Manager, Bitstream FaceLift, and Hewlett-Packard's Intellifont. If any of these font managers is present, they should be disabled or removed from the system for troubleshooting purposes.

Use the Notepad utility to examine the font substitution table `[fontSubstitutes]` in the WIN.INI file. This table determines what Windows will substitute for certain fonts. The values of these entries can be changed by the user through the DOS editor. The substitution table can also be accessed through the Substitution window under the Edit Substitution button of the Printer Setup's Advanced Options dialog box.

The TrueType section of the WIN.INI file should also be checked while the file is opened. Normally the system should be set up so that `TTEnable=1`, `TTifCollisions=0`, and `TTOnly=0`. The 1 enables the system's TrueType fonts; the 0's select TrueType as the first choice, but also allow other fonts to be used in the system.

If printer problems continue, try printing a sample file from a non-Windows environment. The best example of this is to copy the AUTOEXEC.BAT or CONFIG.SYS files to the LPT1 port. If this does not work from the DOS level, a hardware or configuration problem is indicated.

If this is the case, type **EDIT AUTOEXEC.BAT**. Check the file for a `"SET TEMP = "` command. If the command is not present, add a **SET TEMP** statement to the AUTOEXEC.BAT file. At the C:\>> DOS prompt, type **EDIT AUTOEXEC.BAT**. Create a blank line in the file and type **SET TEMP=C:\WINDOWS\TEMP** into it. Save the updated file to disk and reboot the system. Make sure to check the `SET TEMP=` line for blank spaces at the end of the line.

Is there a printer switch box between the computer and the printer? If so, remove the print-sharing equipment, connect the computer directly to the printer, and try to print from the DOS level as previously described.

If the system will print from DOS but not from Windows, determine if the Print option from the application's File menu is unavailable (gray). If so, check the Windows Control Panel/Printers window for correct parallel port settings. Make certain that the correct printer driver is selected for the printer being used.

If no printer or the wrong printer type is selected, just set the desired printer as the Default printer. To add the desired printer as the Default printer, enter the Main window, double-click on the Control Panel icon, double-click on the Printer icon, and set the desired printer as the Default printer.

Click on the Setup button to examine the selected printer's settings. If these settings are correct, click on the Connect button to ensure that the printer information is being routed to the correct port.

If nothing is being produced by the printer even though print jobs have been sent to it, check the Print Spooler to see if a particular type of error has occurred. To view documents waiting to be printed, double-click the desired printer's icon. Return to the Printer folder, right-click the printer's icon, click Properties, and then select Details. From this point, select Spool Settings and select the Print Directly to the printer option. If the print job goes through, there is a spooler problem. If not, the hardware and printer driver are suspect.

To check Spooler problems, examine the system for adequate hard disk space and memory. If the *Enhanced Metafile* (EMF) Spooling option is selected, disable it, clear the spooler, and try to print. To check the printer driver, right-click the printer's icon, select Properties and click Details. Reload or upgrade the driver if necessary.

If a Windows 95 printer operation stalls, or crashes, during the printing process, some critical condition must have been reached to stop the printing process. The system was running but stopped. Restart the system in Safe Mode and try to print again. If the system still will not print, check the print driver, the video driver, and the amount of space on the hard disk drive. Delete backed-up spool files (.SPL and .TMP) in the SYSTEM/SPOOL/PRINTERS directory.

DOS-Based applications should have no trouble printing in Windows 95. Windows 95 has enhanced DOS printing capabilities in that they can take part in the new spooling function and usually results in quicker printing of DOS documents. If a particular DOS application has trouble printing, check other DOS applications to see whether they share the problem. If so, use the normal Windows 95 troubleshooting steps just discussed to locate and correct the problem. If the second DOS application prints correctly, check the print settings of the original malfunctioning application.

Summary

The focus of this chapter has been printers. The opening section of the chapter presented an introduction to the different types of printers and provided a fundamental course in general printer structure and organization.

Following the general discussions of dot-matrix, ink-jet, and laser-printer operations, the chapter focused on troubleshooting procedures for each type of printer. Each procedure was divided into logical areas associated with typical printer symptoms.

The final section of the chapter featured printer problems and solutions associated with the Windows environment.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each point and can perform each task listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercises

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedure 30 - Printer Installation and Setup.

Review Questions

1. List three common pin configurations for dot-matrix printers.
2. Referring to the "Hardware Checks" sections of the troubleshooting procedures for all three printer types, describe three general types of problems common to all printers and the additional type of problem that dot-matrix and ink-jet printers have.
3. Name the four basic components of a laser printer cartridge.
4. What are the common transmission parameters that must be set up for a serial printer interface?
5. Describe the purpose for using pin-feed mechanisms to move paper through the printer.
6. Describe the reason for using tractor-feed paper handling.
7. What is the purpose of the corona wire in a laser printer?
8. If the resolution of an ink-jet printer becomes unacceptable, what action should be taken?
9. Describe the function of the fuser assembly in a laser printer.
10. Describe the function of the primary corona (conditioning roller) in a laser printer.
11. What is the first action that should be taken if the print generated by a dot-matrix printer becomes faded or uneven?
12. If a laser printer continues to show a paper jam problem after the paper has been cleared, what type of problem is indicated, and what action should be taken?

13. What is the first action that should be taken if the print generated by a laser printer becomes faded or uneven?
14. List the three primary areas where paper jams occur in a laser printer, as well as any other areas where jams are likely to occur.
15. Describe two methods used by ink-jet printers to put ink on the page.
16. What type of electrical charge must be placed on the corona wire to transfer toner from the drum to the paper?
17. Does a successful self-test indicate that the printer is not the cause of the problem? List the parts of the system that can still be problem causes if the self-test runs successfully.
18. List the six stages of a typical laser printer.
19. How does a dot-matrix printer actually deliver ink to a page?
20. What functions does the printer's controller typically perform?
21. List the fundamental parts of a dot-matrix printer.
22. List four things that can be damaging to the photosensitive surface of the laser printer's drum.
23. List the basic components of an ink-jet printer.
24. What type of ink delivery system is normally found in ink-jet printers built for the personal computers?
25. Describe what the specification for 60-pound bond paper means.

Review Answers

1. 9, 18, and 24 pins. For more information, see the section titled "Dot-Matrix Characters."

2. Power-supply problems, not printing problems, and paper not advancing problems. The dot-matrix and ink-jet printers may also suffer from printhead not moving problems. For more information, see the sections titled "Troubleshooting Dot-Matrix Printers," "Troubleshooting Ink-Jet Printers," and "Troubleshooting Laser Printers."

3. The toner supply, the corona wire, the drum assembly, and the developing roller. For more information, see the section titled "Component Variations."

4. Speed, parity type, character frame, and control protocol. For more information, see the section titled "Printer Installation."

5. The pin-feed method allows heavier grades of paper to be pulled through the printer. This makes it less likely to slip or become misaligned than with friction-feed paper-handling techniques. For more information, see the section titled "Paper Handling."

6. A tractor feed pulls paper through the printer using the pins in the tractor. The gearing of the tractor makes it especially useful in handling heavy-duty, continuous form paper that tends to be very heavy. For more information, see the section titled "Paper Handling."

7. There are actually two corona wires in a laser printer. The first is the primary corona wire that conditions the drum for printing. The second is the transfer corona that transfers the toner from the drum to the paper. For more information, see the section titled "Laser Printing Operations."

8. Replace the ink cartridge. For more information, see the section titled "Ink Cartridges."

9. The fuser melts the toner on the paper and then presses it into the paper. For more information, see the section titled "Laser Printing Operations."

10. The conditioning roller applies a uniform charge to the surface of the drum to prepare it to receive the next image from the laser. For more information, see the section titled "Laser Printing Operations."

11. Change the ribbon cartridge. For more information, see the section titled "Ribbon Cartridges."

12. An interlock problem has occurred. It will be necessary to open the unit to clear the interlock error. For more information, see the section titled "Paper Will Not Feed, or Is Jammed."

13. Adjust the contrast control and check the toner cartridge. For more information, see the section titled "Print on Page Is Missing or Bad."

14. The pickup area, the registration area, and the fusing area. If optional output devices are included, such as collators and duplexers, jams can occur there as well. For more information, see the section titled "Troubleshooting Laser Printers."

15. Drop-on-demand and continuous-stream. For more information, see the section titled "Ink Delivery Methods."

16. Positive. For more information, see the section titled "Laser-Printing Operations."

17. No. The computer, the printer cable, and the printer's interface circuitry. For more information, see the section titled "Troubleshooting Dot-Matrix Printers."

18. Cleaning, conditioning, writing, developing, transferring, and fusing. For more information, see the section titled "Laser-Printer Components."

19. The designated pins of the printhead are extended from the face of the printhead because of an electrical charge. The pins strike an inked ribbon, which in turn impacts the paper. For more information, see the section titled "Dot-Matrix Characters."

20. It receives data and control signals from the host computer through the interface circuit and generates all the control signals necessary to carry out the operation of the printer as directed. For more information, see the section titled "Printer Controls."

21. The power supply, the interface/controller board, the paper-feed motor and gear set, the printhead mechanism, the printhead-positioning motor and belt, and the sensors. For more information, see the section titled "Dot-Matrix Printers."

22. Light, dust and other particles, high temperature, and high humidity. For more information, see the section titled "Laser-Printing Operations."

23. The power supply, the interface/controller board, the paper-feed motor and gear set, the printhead mechanism, the printhead-positioning motor and belt, and the sensors. For more information, see the section titled "Ink-Jet Printer Components."

24. Drop-on-demand ink delivery. For more information, see the section titled "Ink Delivery Methods."

25. When this paper is stacked in 500 22×17–inch sheets, it will weigh 60 pounds. For more information, see the section titled "Paper Specifications."

Chapter 11

Data Communications

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Define the term modem.
- Define the term baud.
- Describe the three types of modulation commonly used in data communications.
- Explain how baud and bit rate can differ.
- Compare hardware- versus software- (code control) oriented protocols.
- Describe steps to troubleshoot modem problems.
- Describe the operation and hardware of an Ethernet LAN system.
- Describe the operation and hardware of an ARCnet LAN system.
- Differentiate between different types of network media (10base2, 10base5, for example).
- Describe steps to troubleshoot LAN networking problems.
- Describe the function of routers, hubs, and bridges in network systems.

continues

- Discuss basic concepts relating to Internet access (dial-up, ISP connections, browsers, and so on).
- Discuss common Internet concepts and terminology (WWW, e-mail, and so on).
- Describe steps to troubleshoot WAN networking problems.
- Set a system up to use a modem for Internet access via an Internet service provider (ISP).
- Use ISP information to configure PPP and TCP/IP protocols.
- Install and configure a Web browser.
- Work with a LAN administrator to disconnect and reinstall a system from the network without seriously disturbing the network.
- Determine whether a computer is networked or not.
- Exchange network interface cards between similar systems.

Introduction

The most explosive area of personal computer use is in the realm of data communications. Increasingly, personal computers are being connected to one another. Data communications can be as simple as connecting two units together so that they can “talk” to each other. This can be accomplished by wiring their serial or parallel ports together when they are in close physical proximity to each other (up to a few feet). Communicating over longer distances requires additional hardware in the form of a modem or a network card and software, in the form of drivers and protocols.

When more than two computers are linked together so that they can share information, a network is formed. Networks in a relatively confined geographical area are called `local area networks` (LANs); networks distributed over wider geographical areas are referred to as `wide area networks` (WANs).

Modems

Objective

Generally, the most difficult aspect of connecting peripheral equipment to a computer is obtaining the proper interfacing and cabling. If the peripheral is located at some distance from the computer (greater than 100 ft.), it cannot be connected by just getting a longer cable. As the connecting cable gets longer, its resistance combines with distributive capacitance along the wires to form a natural, electrical-signal integrator, which tends to distort digital signals until they are no longer digital.

To overcome this signal deterioration, a device called a `modem` (short for modulator/demodulator) is used to convert the parallel, digital signals of the computer into serial, analog signals that are better suited for transmission over wire. A modem allows a computer to communicate with other computers through the telephone lines, as depicted in Figure 11.1.

Some of the services available through the modem include bulletin board services (BBSs), user groups, and a variety of national and worldwide communication services such as the Internet, Prodigy, and America Online. Many modems incorporate send/

receive `facsimile` (fax) capabilities that allow the computer to correspond directly with fax machines around the world. Some newer modems incorporate digitized, voice-transmission capabilities so that users can employ them as inexpensive telephones.

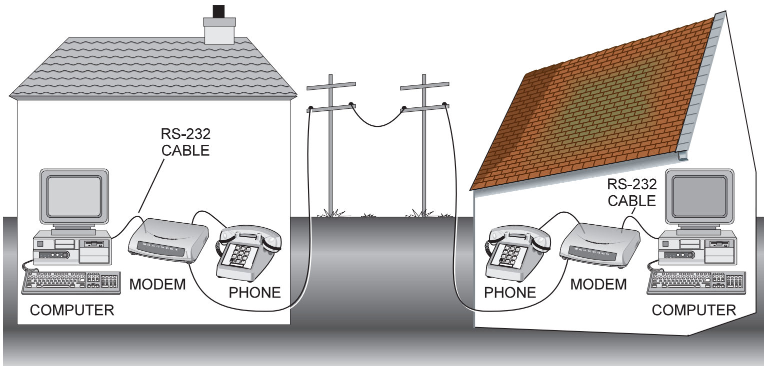


Figure 11.1

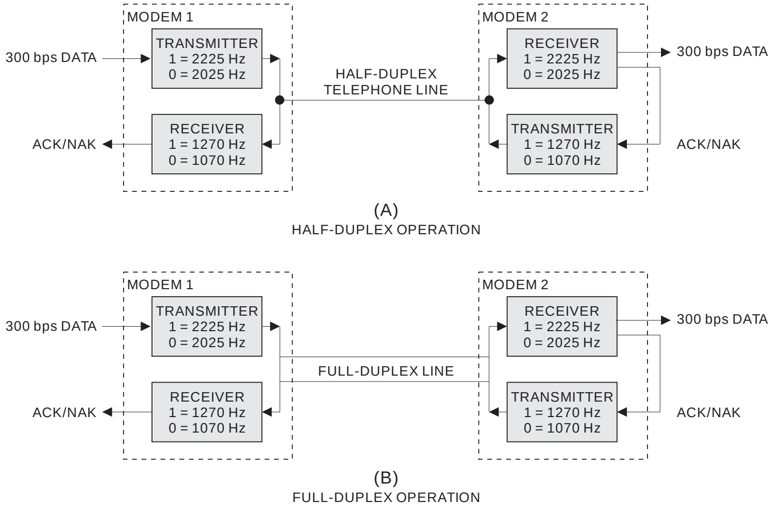
Modem communications.

In its simplest form, a modem consists of two major blocks: a `modulator` and a `demodulator`. The modulator is a transmitter that converts the parallel computer data into a serial format for transmission. The demodulator is the receiver that accepts the serial transmission format and converts it into a parallel format usable by the computer or peripheral.

A modem that sends signals in only one direction, like the one just described, is referred to as a `simplex-mode` modem. In the case of connecting two computers, or a computer and a remote terminal together, it is desirable to transmit data in both directions. This can be accomplished by placing a separate modulator and demodulator at each end of the transmission line. In this configuration, a complete modem unit is present at each end of the line. Modems capable of both transmitting and receiving data are divided into two groups, based on their mode of operation. `Half-duplex mode` modems can exchange data with another modem, but only in one direction at a time, as illustrated in Figure 11.2. Because both modems contain both a modulator and a demodulator, adding an extra conductor will allow both modems to send and receive data simultaneously. This mode of operation is known as `full-duplex mode`.

Figure 11.2

Half- and full-duplex communications.



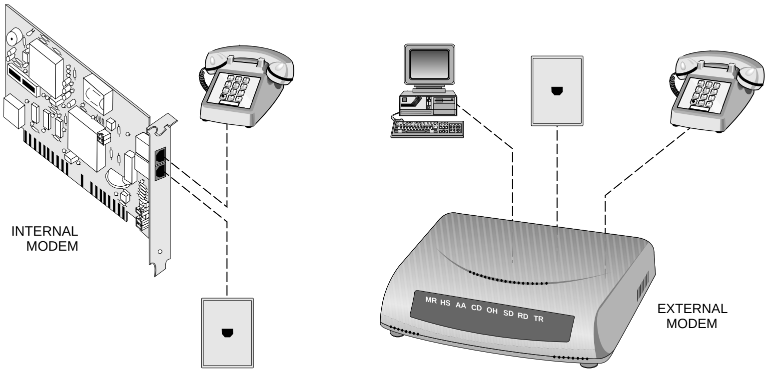
As the distance between terminals increases, it soon becomes impractical to use dedicated wiring to carry data. Fortunately, there is already a very extensive communications network in existence—the public telephone network. Unfortunately, the phone lines were designed to carry analog voice signals rather than digital data. The design of the public phone system limits the frequency at which data may be transmitted over these lines. Once again, the modem can be used to enable computers to use public telephone lines to exchange information.

A modem can be either an internal or an external device, as illustrated in Figure 11.3. An `internal modem` is installed in one of the computer's expansion slots and has its own UART and interfacing circuitry. The `external modem` is usually a box installed outside the system unit and connected to one of the computer's serial ports by an RS-232 cable. An external unit also requires the use of an internal serial port for its UART. Most computers contain two serial port connections.

In both cases, the modem typically connects to the telephone line using a standard 4-pin, RJ-11 telephone jack. A second RJ-11 jack in the modem allows an additional telephone to be connected to the line for voice usage. A still smaller 4-pin, RJ-12 connector is used to connect the telephone handset to the telephone base. Be aware that an RJ-14 jack looks exactly like the RJ-11, but that it defines two lines to allow for advanced telephone features such as Caller ID.

Figure 11.3

Internal and external modems.



Understanding How Modems Work

Objective

The standard telephone system accommodates a range of frequencies between 300 and 3,300 Hz, or a bandwidth of 3,000 Hz. This is quite adequate to transmit voice, but severely distorts digital data. To use the audio characteristics of the phone lines to their best advantage, the modem encodes the digital 1's and 0's into analog signals within this bandwidth.

Modems are generally classified by their `baud rate`. Baud rate is used to describe the number of signal changes that occur per second during the transfer of data. Because signal changes are the quantity actually being limited by the telephone lines, the baud rate is the determining factor. Most modems begin encoding data into different transmission formats so that a number of data bits can be represented by a single signal change. In this way, the bit rate can be high, while the baud rate is still low. Common bit rates for telecommunications include 2,400, 9,600, 14,400, 28,800, and 33,600 bits per second. To complete a successful connection at maximum speed, the other party involved must have a compatible modem, capable of the same baud rate.

FSK Modulation

The most common method of encoding data is a specialized form of `frequency modulation` (FM) called `Frequency-Shift Keying`, or FSK. Using this method, a specific frequency such as 1,070 Hz is used to represent a `0` (or `space`), and a second frequency such as 1,270 Hz is used to represent a `1` (or `mark`). By using a second pair of frequency tones, the modem can both send and receive a message, simultaneously. The sending (or `originating`) modem may use the frequencies shown in the first row in Figure 11.4; the receiving (or `answering`) modem would have to use the frequency scheme shown in the second row.

Figure 11.4

FSK transmission.

		1 MARK (Hz)	0 SPACE (Hz)
ORIGINATE S16=0	TRANSMITTER RECEIVER	1270 2225	1070 2025
ANSWER S16=0	TRANSMITTER RECEIVER	2225 1270	2025 1070
ORIGINATE SELF-TEST S16=1	TRANSMITTER RECEIVER	2225 2225	2025 2025
ANSWER SELF-TEST S16=1	TRANSMITTER RECEIVER	1270 1270	1070 1070

FSK is used in most low-speed, inexpensive modems to transmit data at a rate of 300 bits per second (bps). The terms `baud` and `bps` are sometimes used interchangeably. However, this is true only if a signal change is used to represent each bit. Using FSK modulation, each bit is represented by a signal change, so it is proper to refer to the transfer rate as either 300 bps or 300 baud.

PSK Modulation

At the higher-transmission rates used by medium- and high-speed modems, FSK becomes increasingly ineffective because of signal losses and noise produced by exceeding the bandwidth of the phone line. To allow higher transmission rates, higher-speed modems use a different form of modulation referred to as `phase modulation` or `Phase-Shift Keying` (PSK). PSK modulation encodes data on a sine wave by shifting the phase of the wave to represent the

1's and 0's. In this manner, a single carrier frequency can be used to carry data, with the relative phase of the waveform indicating the bit-value of data. As an example, a logic 0 can be represented as a 0-degrees phase shift, and a logic 1 by a 180-degrees phase shift.

Because phase shift is an analog quantity, more than two values of phase shift can be used to represent data, as indicated in Table 11.1. When using more than two phase relations to encode data, it is referred to as `Differential Phase-Shift Keying` (DPSK). By grouping consecutive data bits together, in groups of two or three, more than one bit of data can be represented by a single quantity of phase shift. Therefore the baud rate can be held within the bandwidth of the transmission line, while the number of bits transmitted is increased. When two bits (`dibits`) are grouped together, there are four possible bit-pattern combinations. These combinations can be represented by four increments of phase shift, as depicted in Figure 11.5. Three-bit groups (`tribits`) produce eight possible bit combinations and require eight distinct phase shifts to encode the data. The most common transmission rate for this type of modem is 1,800 bps, at 600 baud.

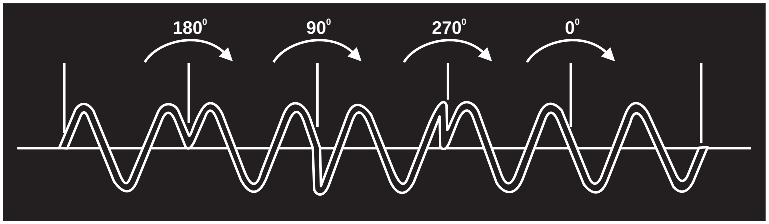


Figure 11.5

Phase-shift keying.

Table 11.1

DPSK encoding.

Dibit Pair	Phase Shift	Tribits	Phase Shift
00	0	000	0
01	90	001	45
10	180	010	90
11	270	011	135
		100	180
		101	225

Dibit Pair	Phase Shift	Tribits	Phase Shift
		110	270
		111	315

QAM Modulation

Even higher transmission rates are achieved by combining DPSK techniques with amplitude modulation (AM), in a method known as quadrature-amplitude modulation (QAM). The combination of the two methods allow two amplitude states and four distinct phase shifts to represent any of the eight possible combinations of a 4-bit group, with a single signal change. Therefore twice as many data bits can be transmitted using quadrature modulation as with PSK alone. In other words, this type of modem commonly transmits data at 2,400 bps, and 600 baud. Using tribits extends the transmission rate to 4,800 bps, at 600 baud.

Data Compression

In addition to modulation techniques, advanced modems use data compression techniques to reduce the volume of data that must be transmitted. These data compression schemes are similar in operation to the techniques used with hard-disk drive recordings, tape backup systems, and the audio/video digitizers described earlier in this text.

Group coding techniques allow redundant patterns of 1's and 0's to be represented by shorter encoded words. Several group coding techniques have been developed. They all work on the same basic premise, however, removing redundant information from the data flow so that it takes less time to transmit. Each method involves a mathematical algorithm that reads the data and converts it into encoded words. The modem at the receiving end must use the same algorithm to decode the words and restore them to their original form.

Some modem compression standards reach ratios as high as 4 to 1. The major standards for modem data compression have come from a company named Microcom and the Comité Consultatif

International Télégraphique et Téléphonique (CCITT) worldwide standards organization. The Microcom Networking Protocol level 7 (MNP 7) standard can produce 3:1 compression ratios; the CCITT V.42bis standard reaches 4:1.

Transmission Synchronization

Data transmissions among computers or among computers and peripherals occur in data blocks consisting of strings of characters. There are two techniques of synchronizing the timing of data transmission among devices. The first method, known as synchronous transmission, places timing synchronization signals at the beginning of each data block to synchronize the transmitter and the receiver. Because the entire block of data is synchronized at the beginning, blocks of data may be transmitted, one block after another, at a maximum rate set by the transmission line and the connecting modems. Because the entire block of data, the timing signals, and error-checking information must be assembled before transmission, this technique requires the direction of a microcontroller and is usually reserved for high-speed, intelligent modem systems.

The other technique, called asynchronous transmission, places a synchronizing bit at the beginning and end of each 8-bit character. These bits are referred to as start and stop bits, and allow characters to be transmitted at irregular intervals. This mode of transmission is usually associated with lower-speed modems (those in the range used with microcomputers and PCs).

Modem Sophistication

The simplest modem operation is provided by manually dialed modems. With this type of modem, an operator at one terminal dials a telephone number to contact an operator at another terminal. After the connection has been made, the operators initiate data transfers between their respective computers. At the end of the transmission, the operators must terminate the connection.

There are typically a number of sophisticated functions available with more intelligent modems. Among these features is autodialing. Autodial modems allow lists of telephone numbers to be stored under the control of software. These numbers can be dialed by the system at some time specified by you (such as at night when telephone rates are cheaper), and may include automatic redialing of busy numbers. Besides autodial, there are also modems that provide autoanswer, auto logon, and auto disconnect functions to allow completely unattended transfers of data to take place. These are commonly referred to as originate/answer modems, because of their capability to both initiate and answer calls under program control.

Several features have been added to the basic modem that go far beyond simple autodial/autoanswer capabilities. The first major feature addition is the capability to communicate with facsimile machines (faxes). Many newer fax/modems come with the capability to send information to a remote fax machine or to receive data from a fax machine and store it on the hard drive until it can be printed out.

Another interesting feature being added to modems is the capability to also carry digitized voice. A voice modem communicates with other modems, but it can also digitize voice and communicate it to another voice modem. These are becoming particularly interesting when used over the Internet. Because of the Internet's organizational structure, voice modems can carry out telephone-like communications between any points in the world, often at the same cost as a local telephone call.

Computer-Modem Transfers

To facilitate the transfer of data from the parallel bus of the computer to the serial bit stream used by the modem (and vice versa), a UART or USART is used. Most modems use a UART. When the computer sends a character to the UART, it is loaded into a character-buffer register. The UART shifts a start bit into the buffer along with the proper error-checking bit and a specified number of stop bits. The character data block is now ready for transmission.

At the receiver, the start bit of the incoming character block is detected. This causes the bits that follow to be shifted into the serial-shift register. The error-checking bit is compared to an error bit generated by the receiver, as the character block was shifted into the register. The start and stop bits are stripped from the character block, leaving only the data in the character buffer. The system processor can now access the incoming character in parallel form.

The UART and a number of other asynchronous support chips make up the computer/modem interface. The physical location of the interface, and its complexity, are dependent on the particular computer system and the type of modem being used.

`Standalone modems` do not require an onboard UART because there is one present in the computer's asynchronous adapter card or serial interface. On the other hand, `dedicated modems` require an onboard UART to facilitate direct communications between the computer and the modem.

Protocols

Objective

To maintain an orderly flow of information between the computer and the modem, and between the modem and another modem, a `protocol`—a set of rules governing the transfer of information—must be in place. As far as standalone modems are concerned, the most basic hardware standard is the RS-232C serial interface standard. But within the realm of the RS-232 standard, a proliferation of communication methods exist.

RS-232C

The RS-232C standard identifies communication equipment using two categories:

`Data Terminal Equipment` (DTE), usually a computer

`Data Communication Equipment` (DCE), usually a modem

Data terminal equipment is any equipment whose main purpose is to process data. On the other hand, any communication equipment that changes data during transmission is referred to as DCE. Figure 11.6 illustrates a typical DTE/DCE relationship.

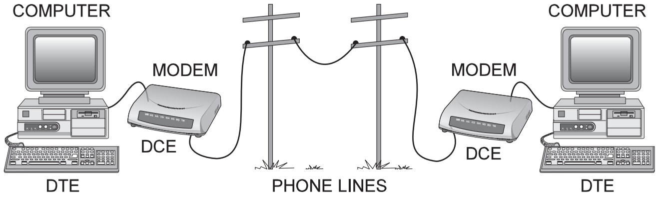


Figure 11.6

The DTE/DCE relationship.

The RS-232C serial interface connection is usually made with a 25-pin, D-type connector. You should be aware that not all RS-232C connections use the 25-pin, DB connector; and not all 25-pin, DB connectors are RS-232C connections. Table 11.2 describes the pin designations and functions associated with the RS-232C interface, as they apply to modem/computer connections.

Table 11.2

RS-232C signal/pin definitions.

Pin Number	Common Name	RS-232C Name	Description
1		AA	Protective ground
2	TXD	BA	Transmitted data
3	RXD	BB	Received data
4	RTS	CA	Request to send
5	CTS	CB	Clear to send
6	DSR	CC	Data set ready
7	GND	AB	Signal ground (common return)
8	CD	CF	Received line signal detector(RLSD)
9			Reserved for data set testing
10			Reserved for data set testing
11			Unassigned
12	SI	SCF	Secondary rec'd line sig. detector

continues

Table 11.2 Continued

Pin Number	Common Name	RS-232C Name	Description
13		SCB	Secondary clear to send
14		SBA	Secondary transmitted data
15		DB	Transmission signal element timing
16		SBB	Secondary received data
17		DD	Receiver signal element timing
18			Unassigned
19		SCA	Secondary request to send
20	DTR	CD	Data terminal ready
21		CG	Signal quality detector
22	RI	CE	Ring indicator
23		CH/CI	Data signal rate selector
24		DA	Transmit signal element timing
25			Unassigned

In its most basic form, the RS-232C interface makes provision for full-duplex operating mode through pin 2 (TXD) and pin 3 (RXD). Normally, data passes from the DTE to the DCE on the TXD (transmitted data) line, and from the DCE to the DTE on the RXD (received data) line, although these two pins may sometimes be reversed.

Of the variations of protocol schemes in use with the asynchronous RS-232C standard, five types are in widespread use today. These five types of protocols can be broken down into two distinct classes:

- Hardware-oriented protocols
- Control-code–oriented protocols

Hardware-Oriented Protocols

The hardware oriented protocols are tied to the use of a particular pin of the RS-232C connector to control data flow. The two most common forms of hardware protocols are DTR and RTS, named after the `data terminal ready` and `request to send` pins. These lines are toggled between high- and low-logic levels to control when to send and not send data. The DTE uses the RTS (pin 4) to inform the DCE that it is ready to send data. The DTE may also use the DTR (pin 20) to inform the DCE that it is online and functioning.

The DCE uses a trio of reciprocal lines, CTS (`clear to send`), DSR (`data set ready`), and CD (`data carrier detect`) to signal the Data Terminal Equipment. It uses the CTS to inform the DTE that it is ready to accept data. The data set (or modem) uses the DSR to notify the DTE that it is connected to the phone line.

The RS-232C standard also designates a number of other lines that can be used for specialized functions. The `speed indicator` (SI) line connected to pin 12 is used by the DCE to indicate whether the modem is in low- or high-speed mode. The DCE may also use the `ring indicator` (RI) line to indicate that ring-in voltage is being received. Pin 1 of the DB-25 connector is reserved for the `protective frame` ground; pin 7 is used for the `signal ground`, to which all other voltages in the interface are referenced.

Control-Code–Oriented Protocols

Most data-flow control is performed using the control-code, or software, class of protocols. Of this class of protocols, three types are in widespread use:

- XON/XOFF
- ACK/NAK
- ETX/ACK

In these protocols, control codes are sent across the data lines to control data flow, as opposed to using separate control lines. Table 11.3 presents a listing of accepted ASCII control codes.

Table 11.3

Control codes.

Hex	EBCDIC	ASCII	Binary	Description
00	NUL	NUL	0000 0000	Character (used for padding)
01	SOH	SOH	0000 0001	Start of header (begin session)
02	STX	STX	0000 0010	Start of text (begin data block)
03	ETX	ETX	0000 0011	End of text (end data block)
04	PF	EOT	0000 0100	End of transmission
05	HT	ENQ	0000 0101	Enquire (bid for acknowledgment)
06	LC	ACK	0000 0110	Acknowledge (positive response)
07	DEL	BEL	0000 0111	Bell (ring the bell)
08	GE	BS	0000 1000	Back space
09	RLF	HT	0000 1001	Horizontal tab
0A	SMM	LF	0000 1010	Line feed
0B	VT	VT	0000 1011	Vertical tab
0C	FF	FF	0000 1100	Form feed
0D	CR	CR	0000 1101	Carriage return
0E	SO	SO	0000 1110	Shift out (subscripting)
0F	SI	SI	0000 1111	Shift in (superscripting)
10	DLE	DLE	0001 0000	Data link escape (binary xmission)
11	DC1	DC1	0001 0001	Device control #1 (XON)
12	DC2	DC2	0001 0010	Device control #2

Hex	EBCDIC	ASCII	Binary	Description
13	TM	DC3	0001 0011	Device control #3 (XOFF)
14	RES	DC4	0001 0100	Device control #4
15	NL	NAK	0001 0101	Not acknowledge
16	BS	SYN	0001 0110	Synchronous idle (sync character)
17	IL	ETB	0001 0111	End-of-transmission block
18	CAN	CAN	0001 1000	Cancel
19	EM	EM	0001 1001	End of medium
1A	CC	SUB	0001 1010	Substitute character
1B	CU1	ESC	0001 1011	Escape
1C	IFS	FS	0001 1100	Field separator
1D	IGS	GS	0001 1101	Group separator
1E	IRS	RS	0001 1110	Record separator
1F	IUS	US	0001 1111	Unit separator
20	DS	SP	0010 0000	Space (blank character)

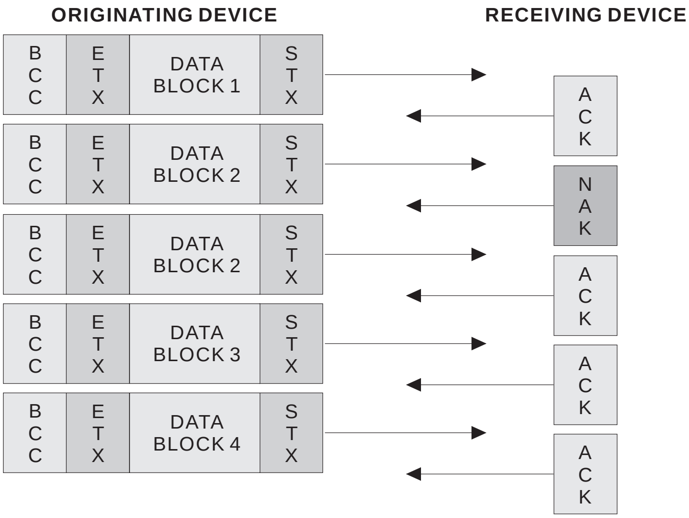
The XON/XOFF protocol, where X represents two special control characters, is a relatively simple concept used to regulate data flow. This control is necessary to prevent buffer memories from overfilling. When data overflows the buffer, the result is usually an error code. The XON/XOFF protocol uses special control characters to start and stop data flow.

The ACK/NAK and ETX/ACK protocols are considered to be "high-level" protocols because they require special interface programs called `drivers` or `driver emulators`. In both cases, these protocols use special control characters and escape-code sequences to provide functions such as data transmission integrity, flow control, requests for retransmission, and so forth.

The ACK/NAK protocol derives its title from the ASCII control characters for ACKnowledge, and Not ACKnowledge. It uses these characters to provide a means of error correction for transmitted data. Basically, the ACK/NAK protocol expects a block of data to be preceded by a `start-of-text` (STX) character, and to be followed by both an `end-of-text` (ETX) character and an `error-checking code` (BCC), as depicted in Figure 11.7. At the receiving end, the block-check character (BCC) is checked for errors. Depending on the outcome of the check, either an ACK signal will be returned, indicating a successful transmission, or a NAK signal, indicating an error has occurred. If a NAK signal is returned, the transmitting device responds by retransmitting the entire block.

Figure 11.7

ACK/NAK transmission.



The ETX/ACK protocol is somewhat simpler than ACK/NAK, in that no character check is performed. If the receiving device does not return an ACK signal within a predetermined length of time, the sending device assumes an error or malfunction has occurred, and retransmits the character block.

The CCITT and MNP protocols for data compression were mentioned earlier in this chapter. Actually, there are many protocol standards in use with data communications equipment.

The CCITT standards are identified by a v.xx nomenclature. The original CCITT standard was the v.22 protocol that established transfers at 1,200 bps using 600 baud. The v.22bis standard followed, providing 2,400 bps transfers at 600 baud using QAM methods. The v.32 protocol increased the bps rate to 4,800 and 9,600. The 4,800 bps rate uses QAM modulation at 2,400 baud; the 9,600 bps rate operates at 2,400 baud with additional QAM bit values. A v.32bis modification improves the transmission rate to 14,400 bps by providing 128 possible QAM values.

The CCITT standards also include an error-correction and a data-compression protocol. The v.42 standard is the error-correction protocol; v.42bis protocol is the CCITT equivalent of the MNP5 and MNP7 protocols. Both protocols run as modules along with the v.32 and v.32bis protocols to provide additional transmission speed.

The MNP Microcom standards began with protocols MNP2 through MNP4. These standards dealt with error-correction protocols. The MNP5 and MNP 7 standards followed as the first data-compression protocols. The MNP10 standard introduced the first `Adverse Channel Enhancement` protocol. This type of protocol is designed to provide maximum performance for modems used in typically poor connection applications such as cellular phones. It features multiple connection attempts and automatically adjusted transmission rates. Like the advanced CCITT protocols, the MNP10 protocol module runs along with a v.42 protocol to maximize the data transmission rate.

Newer CCITT and MNP protocols are under development to provide modems with 56Kbps transmission capabilities. These protocols use a two-step mapping algorithm to map 256 binary values into a data encoder. The encoder produces digital symbols that are applied to a D/A circuit that converts it into 256 possible analog signals. The symbol rate for 56Kbps modems is 8KHz. The two-step mapping algorithm provides the best possible trade-off between data table size, error rates, and elevated complexity.

During a special `training` period conducted at lower speeds, the modem tests the integrity of the transmission medium. The modem uses the training information to determine the ultimate group of points in the map to be used for the existing line conditions. The modem then negotiates with the remote modem to determine the maximum transfer rate.

Several error-correcting file transfer protocols have been developed for modem communications packages. Some of the more common protocols include the following:

- Xmodem
- Ymodem
- Zmodem
- Kermit

These protocols use extensive error-detection schemes to maintain the validity of the data as it's being transmitted. The concept of parity checking has already been discussed as a method of detecting data errors. However, parity checking alone is somewhat ineffective when used to check large blocks of data. It also offers no convenient means of error correction.

Error-detecting and correcting protocols generate more exotic error-detection algorithms such as `checksum` and `cyclic redundancy checks` (CRCs) to identify, locate, and possibly correct data errors.

In `Xmodem` transfers, the transmitter sends a data block like the ones depicted in Figure 11.7. The data section of the block is 128 bytes long. A checksum character is used to detect transmission errors and the ACK/NAK method of flow control is employed. An improved version of the protocol, called `Xmodem CRC`, replaces the checksum character with a CRC block-check character.

The `Ymodem` protocol improves on the Xmodem CRC version by increasing the size of the data block to 1,024 bytes. Ymodem can still support the 128-byte Xmodem format, if necessary. The 1,024-byte version begins with an STX ASCII character; the 128-byte transmission starts with an SOH character. An improved version, called `Ymodem batch`, includes filenames and sizes in the transfer to allow multiple files to be sent in a single transmission. A further improvement is used in `Ymodem G`. In this format, the sending unit uses the Ymodem batch method, but does not wait for an ACK signal back from the receiver before sending the next data block. This last format is used only along with the MNP protocol functions that supply their own error checking and correcting.

The `Zmodem` protocol provides a wealth of high-level management features. In addition to using CRC16 and CRC32 error-detection schemes to verify data integrity, the Zmodem protocol offers Auto-file Restart crash-recovery techniques and selective file transfers. The files are selected by including their filenames in a batch transfer operation. Zmodem can be used for transferring both text and binary files (such as EXE files).

All the protocols mentioned in this category are used to transmit files over dial-up telephone lines. However, they are not capable of being used with Internet communications.

It should be pointed out that these are only classes of protocols, and that within these classes, there are many methods of actually implementing a particular protocol. Within a particular protocol, a number of parameters must be agreed on before an efficient exchange of information can occur. Chief among these parameters are `character-type` and `character-framing`. Basically, character-type refers to the character set, or alphabet, understood by the devices. Depending on the systems, the character set may be an 8-bit, ASCII line code; a 7-bit, ASCII code (with a parity bit); or an EBCDIC code.

Character-framing, on the other hand, refers to the total number of the bits used to transmit a character. This includes the length of the coded character and the number and type of overhead bits required to send the character. A common character-framing scheme calls for a start bit, seven data bits, an odd-parity bit, and a stop bit.

Although this is a typical character-framing technique, it is not universal throughout the industry. The problem here is one of device comprehension. The local unit may be using a 10-bit character frame consisting of a start bit, seven data bits, an odd-parity bit, and a stop bit. If the remote system is using something besides 7-bit, odd-parity ASCII, with one stop bit, however, the response from it would be unintelligible as anything written in English.

The Call

To understand the operation of the modem, it is important to understand that it must function in two different modes. The first mode is the `local command state`. In this condition, the modem is "offline" and communicates with the host system to receive and process commands.

The second mode is the `online state`. In this condition, the modem facilitates the transfer of data between the host computer and a remote computer or device. Two events can cause the modem to shift from the offline state to the online condition. The system can prompt the modem to go online and dial out to another unit. To accomplish this, the host computer places the modem in Originate mode. The second event involves the modem receiving a "ring" signal from a remote device. In this situation, the host system shifts the modem into Answer mode.

The modem will automatically shift from the online state to the local command state whenever the carrier signal from the incoming line is not detected in a given amount of time, or is lost after the connection has been made. An embedded code in the transmitted data can also be used to shift the modem into the local command state.

There are basically three conditions to be understand in the operation of the modem:

- Dialing
- Data exchange (the call)
- Answering

Dialing

To place a call using the modem, the same series of events must occur as when you are placing a voice telephone call. When the handset is removed from its cradle, a switch inside the telephone closes, connecting the phone to the line. At this point, the phone is in a state known as `off-hook`, but it is not yet online. When the modem's relay closes, and the dial tone is detected, the modem notifies the host computer that it is connected to the line.

To originate a call using an autodial modem, you must first place your modem in Originate mode. This requires that it be operating in the local command state. Basically, the modem is always in one of two states: the local command state or the online state. In the local command state, you can issue commands from the computer to the modem. In the online state, communications between modems occur.

Using an autodial modem, the modem automatically places the call by issuing the digital tones equivalent of the desired phone number. The number may come from the keyboard, or it may be one that has previously been entered into memory. Some autodial modems are capable of producing both pulse- and touch-tone dialing equivalents.

A number of things can occur to prevent the modem from going into the online state. An intelligent modem will wait a specified length of time after pick-up before it starts dialing. This allows the phone system time to apply a dial tone to the line. After the number has been dialed, the modem waits for the ringback from the telephone company (this is what you hear when you are making a call). When the ringing stops, indicating that the call has gone through, the modem will wait a specified length of time for an answer tone (carrier) from the receiving modem. If the carrier is not detected within the allotted time, the originating modem will begin automatic disconnect procedures (hangs up), as depicted in Figure 11.8. If a busy signal is detected, the originating modem will hang up or refer to a second number.

The Conversation

After the phone line connection has been established, a `handshaking sequence` occurs. The originating modem's carrier received pin (CD) signals its computer that the receiving modem is online. This is followed by a signal on the clear to send (CTS) pin, indicating that its computer is ready to transmit and receive data. The originating modem responds by issuing its own carrier tone frequency, which the answering modem must detect to notify its computer (through its CD and CTS pins) that it is online and ready for data. If both tones have been received, the greetings and handshakes are completed, and the transfer of information begins.

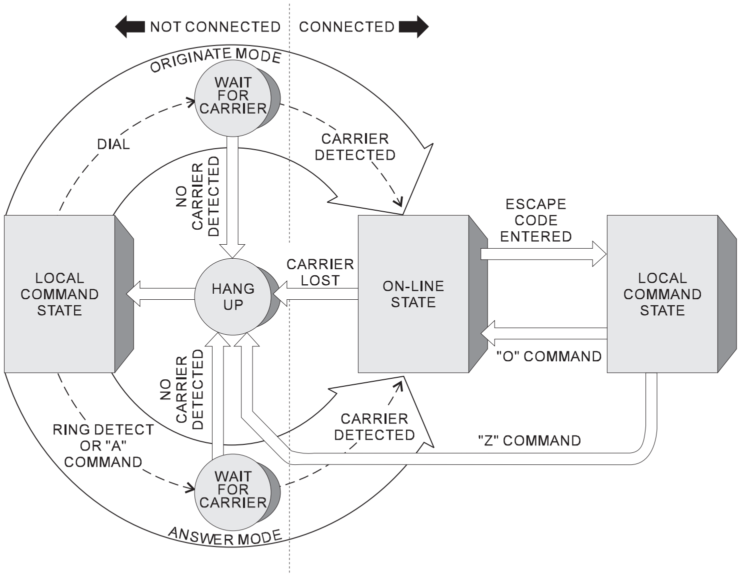


Figure 11.8

Autodial/ autoanswer modem cycle

During the data transfer, both modems monitor the signal level of the carrier to prevent the transfer of false data due to signal deterioration. If the carrier signal strength drops below some predetermined `threshold level`, or is lost for a given length of time, one or both modems will initiate automatic disconnect procedures.

While the modem is in the online state, no commands can be given to it from the keyboard. However, the local command state can be re-entered while still maintaining the connection by use of an escape-code sequence.

Answering

For an autoanswer modem to receive calls, it must be configured to answer the incoming call. When a call comes in, the ring voltage is detected by the modem, which activates its ring indicator (RI) pin to notify the computer that a call is coming in. Depending on the configuration of the modem, it may answer the incoming call on the first ring, or after some preset number of rings. If the called computer is ready, meaning the data terminal ready (DTR) signal is active, the modem goes off-hook and begins the handshaking routine.

The Serial Interface

In the case of external modems, the serial interface, including the UART, is normally provided by the host system. In pre-Pentium units, the system's Multi I/O card provided a pair of fully programmable, asynchronous communication channels through two serial port connections. On earlier I/O cards, a pair of 8250 UARTs were used as the basic port circuitry. In newer MI/O cards, a single VLSI device, called an integrated I/O controller, provided the interfacing and UART functions. In most Pentium systems, the serial port adapter function is incorporated into the system board's integrated I/O controller IC.

The original serial adapters featured programmable baud rates from 50 to 9,600 baud, a fully-programmable interrupt system, and variable character lengths (5-, 6-, 7-, or 8-bit characters). In addition, the adapter added and removed start, stop, and parity bits, had false start-bit detection, line-break detection and generation, and possessed built-in diagnostics capabilities. As modems became faster and faster, upgraded UARTs were included, or integrated, to keep up.

Notable advanced UART versions include the 16450 and 16550. The 16450 was the 16-bit improvement of the 8250; the 16550 was a high-performance UART, with an onboard 16-byte buffer. The buffer allows the UART to store or transmit a string of data without interrupting the system's microprocessor to handle them. This provides the 16550 with an impressive speed advantage over previous UARTs. These advanced UARTs allow serial ports to reach data transmission rates of up to 115 Kbps. Although some features have changed between these UARTs, and although they are sometimes integrated directly into an integrated I/O chip, they must still adhere to the basic 8250 structure to remain PC-compatible.

In most PC-compatible serial ports, very little circuitry besides the UART is needed to implement the interface. Normally, only an oscillator/clock circuit and some RS-232 line driver/receiver chips are required. Figure 11.9 shows these required sections in block form.

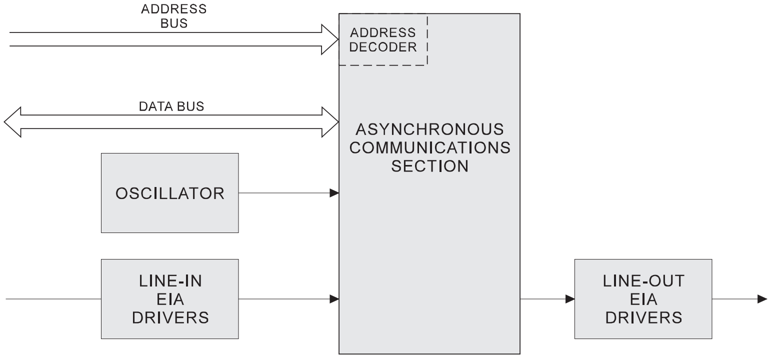


Figure 11.9

RS-232C serial interface.

Although the basic lines of the serial ports originate on a Multi I/O card, or system board, they must be connected to RS-232–compatible, D-shell connectors to complete the interface. This usually involves connecting a ribbon cable between the board that contains the interface circuitry and the connectors, located on a special expansion slot cover. The connectors on the slot cover are usually a 9-pin, DB-9M, and a 25-pin, DB-25M. The ribbon cables pass the TXD, RXD, DSR, DTR, RTS, CTS, CD, RI, and ground signals to the connectors.

As mentioned earlier, normally two serial ports are provided in a PC-compatible system:

- RS-232-1
- RS-232-2

These are hardware settings for the physical ports. The system recognizes the serial ports by their `COM port settings`. COM port settings are just port addresses (COM1, COM2, COM3, and COM4) assigned to the serial interfaces by the operating system. These settings allow the hardware and software to work together. The communication protocol is a function of the system software, and must be loaded before the interface can be operational.

Software uses the COM setting to address instructions and data to the correct UART. For the interface to function properly, therefore, both the hardware and software COM settings must agree. Either RS-232 port may be designated as COM1, COM2, COM3, or COM4, as long as both ports are not assigned to the same COM port number. In most PCs, COM1 is assigned as port address hex 3F8h, and COM2 is assigned port address hex 2F8h. Normally, IRQ4 is selected for COM1, and IRQ3 is selected for COM2. Because these COM ports share IRQ settings, it should be apparent that all four ports cannot be in use at the same time. For example, if an external modem is connected to COM2, then no other device should be set up on COM4.

The UART

Whether the UART is located on a Multi I/O board, in an integrated I/O controller chip, or is part of an internal modem, it must supply most of the logic and control circuitry for the RS-232C interface. In addition to the classic functions described earlier for a UART, the I/O controller provides all the control and handshake lines required for asynchronous communications to take place (that is, RI, DSR, DTR, RTS, CTS, and RLSD). Recall that the receive line signal detect (RLSD) line can also be described as the carrier detect (CD) line, in the RS-232C standard. Of course, these lines are used by the interface adapter to control the flow of information between the PC and the serial communications equipment. Figure 11.10 shows a block function diagram of an 8250-compatible UART.

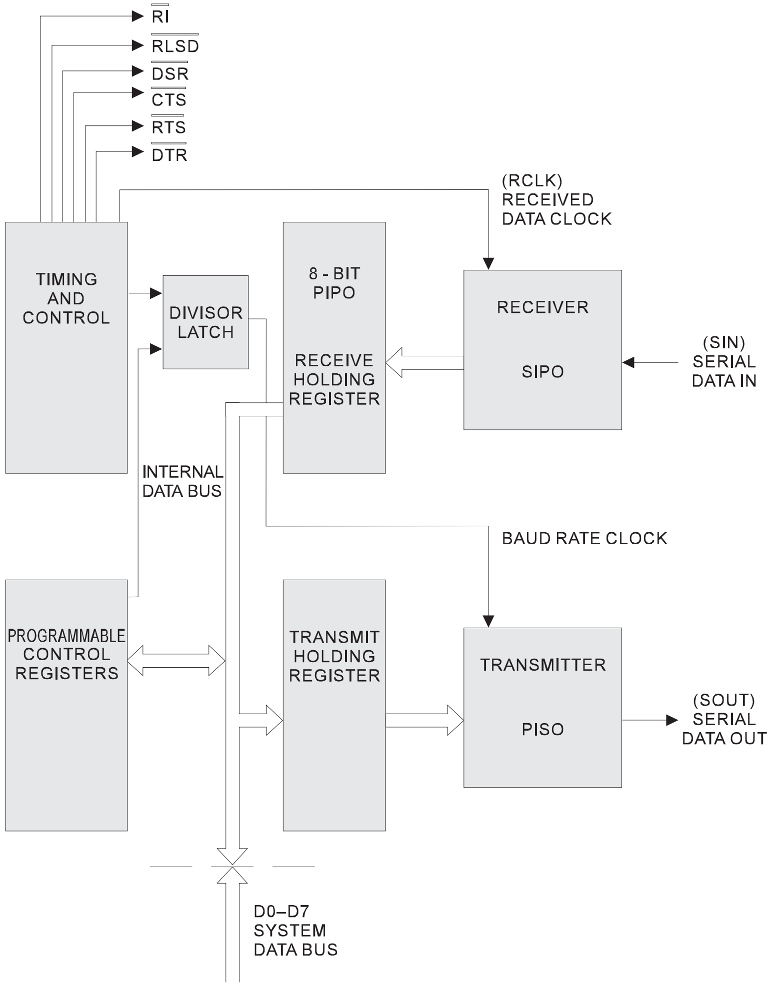
The UART in a PC-compatible system has a number of addressable registers that allow the system unit's software to control its operation. Different modes of operation are selected by programming its registers at port addresses 3F8h through 3FFh for the primary RS-232-1 interface, and 2F8h through 2FFh for the secondary RS-232-2 interface.

The operation of these registers may be summarized as follows:

> TX-RD buffer (3F8/2F8h): This address selects the received data register, which holds the received character, if the UART is in read configuration. Conversely, the transmitter holding register is accessed by the same address when the UART is in write configuration and bit-7 is low. If bit-7 is high, this address accesses the least-significant byte (LSB) of the programmable baud-rate generator for initialization.

Figure 11.10

Inside the UART.



Interrupt enable register (3F9/2F9h): The bits of this register enable four different UART interrupts, which in turn activate the chip's INTRPT output. This address also depends on bit-7 of the line-control register. The Interrupt Enable Register is selected by bit-7 being low. If bit-7 is high, this address selects the most-significant byte (MSB) of the programmable baud-rate generator for initialization.

Interrupt identification register (3FA/2FAh): The contents of this register are used to prioritize four interrupt levels in the UART. The bits of the register are used by software to identify the highest-priority interrupt that may be pending, and lock out all other interrupts, until that particular interrupt has been serviced.

Line control register (3FB/2FBh): The contents of this register are used to define the format of the port's data exchange. The logic level of the various bits of this register define parameters such as word length, number of stop bits, and the type of parity used.

Modem control register (3FC/2FCh): The bits of this register control the DTE/DCE interface (in particular, the DTR and RTS lines). One bit of this register is used by software to provide loopback diagnostic testing.

Line status register (3FD/2FDh): This register provides the system unit with status information concerning data transfers. This information includes data ready in the receiver register, receiver register overrun errors, parity errors, framing errors, transmitter holding register empty, and transmitter shift register empty.

Modem status register (3FE/2FEh): The contents of this register indicate the status of the DTE/DCE control lines. These lines include the CTS, DSR, RI, and RLSD lines.

Serial Cables

Although the information in Table 11.2 shows a designation for nearly every pin in the RS-232 connection (except 11, 18, and 25), many of the pins are not actually used in most serial cables. Figure 11.11 illustrates the basic 25-pin–to–25-pin variation of the RS-232 serial cable. In this example, the connection depicted is a straight-through cabling scheme associated with PCs and PC-XTs.

Because of the advent of the PC-AT, the system's first serial port has typically been implemented in a 9-pin, D-shell male connector on the DTE. Figure 11.12 depicts a typical 9-pin–to–25-pin connection scheme. Notice the crossover wiring technique employed for the TXD/RXD lines displayed in this example. This type of connection became popular with the 9-pin, PC-AT serial port.

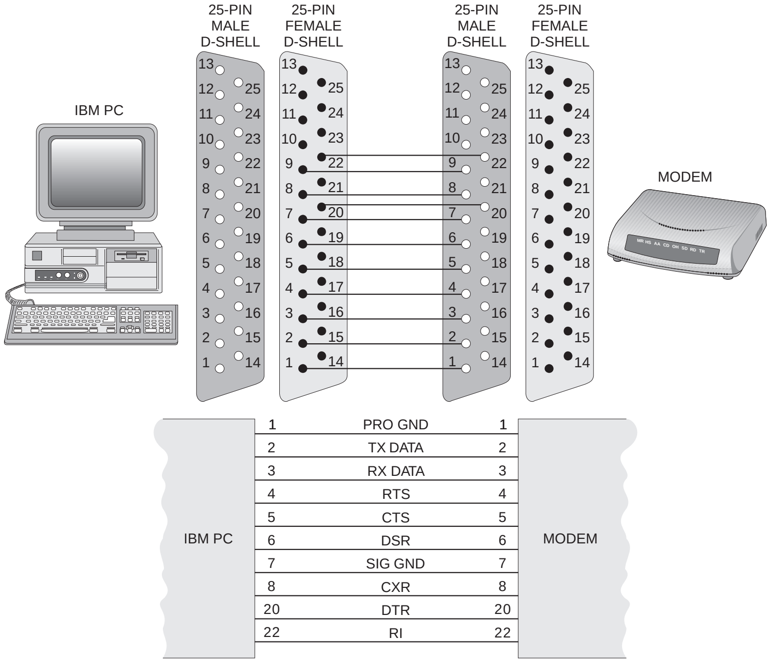


Figure 11.11

A 25-pin–to–25-pin RS-232 cable.

In cases where the serial ports are located close enough to each other, a `null modem` connection can be implemented. A null modem connection allows the two serial ports to communicate directly without using modems.

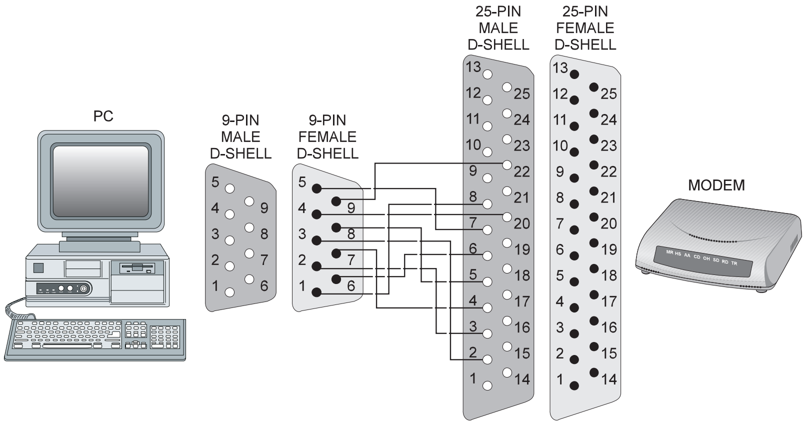
In any event, it should be apparent from the previous trio of figures that all serial cables are not created equal. Incorrect serial cabling can be a major problem when attaching third-party communication equipment to the computer. Read the DCE User's Manual carefully to make certain the correct pins are being connected together.

Communication Software

All modems require software to control the communication session. This software is typically included with the purchase of the modem. At the fundamental-instruction level, most modems use a set of commands known as the `Hayes-Compatible Command Set`. This set of commands is named after the Hayes Microcomputer Products company that first defined them. The command set is based on a group of instructions that begin with a pair of attention characters, followed by command words. Because the attention characters are an integral part of every Hayes command, the command set is often referred to as the `AT Command Set.`

Figure 11.12

A 9-pin–to–25-pin RS-232 cable.



PC pin	Signal	MODEM pin
3	TX DATA	2
2	RX DATA	3
7	RTS	4
8	CTS	5
6	DSR	6
5	SIG GND	7
1	CXR	8
4	DTR	20
9	RI	22

In the Hayes command structure, the operation of the modem shifts back and forth between a `command mode` (local mode), and a `communications mode` (remote mode). In the command mode, the modem exchanges commands and status information with the host system's microprocessor. In communications mode, the modem facilitates sending and receiving data between the local system and a remote system. A short `guard period` between communications mode and command mode allows the system to switch smoothly, without interrupting a data transmission.

AT Command Set

Hayes-compatible AT commands are entered at the command line using an ATXn format. The Xn nomenclature identifies the type of command being given (X), and the particular function to be used (n). Except for ATA, ATDn, and ATZn commands, the AT sequence may be followed by any number of commands. The ATA command forces the modem to immediately pick up the phone line (even if it does not ring). The Dn commands are dialing instructions, and the Zn commands load different default initialization information into the modem. Table 11.4 provides a summary of the Hayes-compatible AT command set.

Table 11.4

The AT Command Set summary.

Command		Function
A/		Re-execute command.
A		Go off-hook and attempt to answer a call.
B0		Select v.22 connection at 1,200 bps.
B1	*	Select Bell 212A connection at 1,200 bps.
C1	*	Return OK message.
Dn		Dial modifier (see dial modifier).
E0		Turn off command echo.
E1	*	Turn on command echo.
F0		Select autodetect mode (equivalent to N1).
F1	*	Select v.21 of Bell 103.
F2		Reserved.
F3		Select v.23 line modulation.
F4		Select v.22 or Bell 212A 1,200 bps line speed.
F5		Select v.22bis 7200 modulation.
F6		Select v.32bis or v.32 4,800 line modulation.
F7		Select v.32bis 7200 line modulation.

Command		Function
F8		Select v.32bis or v.32 9600 line modulation.
F9		Select v.32bis 12000 line modulation.
F10		Select v.32bis 14400 line modulation.
H0		Initiate a hang-up sequence.
H1		If on-hook, go off-hook and enter command mode.
I0		Report product code.
I1		Report computed checksum.
I2		Report OK.
I3		Report firmware revision, model, and interface type.
I4		Report response.
I5		Report the country-code parameter.
I6		Report modem data pump model and code revision.
L0		Set low speaker volume.
L1		Set low speaker volume.
L2	*	Set medium speaker volume.
L3		Set high speaker volume.
M0		Turn speaker off.
M1	*	Turn speaker on during handshaking and turn speaker off while receiving carrier.
M2		Turn speaker on during handshaking and while receiving carrier.
M3		Turn speaker off during dialing and receiving carrier and turn speaker on during answering.
N0		Turn off automode detection.
N1	*	Turn on automode detection.
O0		Go online.
O1		Go online and initiate a retrain sequence.
P		Force pulse dialing.

continues

Table 11.4 Continued

Command		Function
Q0	*	Allow result codes to PC.
Q1		Inhibit result codes to PC.
Sn		Select S-Register as default.
Sn?		Return the value of S-Register n.
=v		Set default S-Register to value v.
?		Return the value of default S-Register.
T		Force DTMF dialing.
V0		Report short form (terse) result codes.
V1	*	Report long form (verbose) result codes.
W0	*	Report PC speed in EC mode.
W1		Report line speed, EC protocol, and PC speed.
W2		Report modem speed in EC mode.
X0		Report basic progress result codes, OK, Connect, Ring, No Carrier (also, for busy, if enabled, and dial tone not detected: No Answer and Error).
Xl		Report basic call progress result codes and connections speeds, that is, OK, Connect, Ring, No Carrier (also, for busy, if enabled, and dial tone not detected: No Answer, Connect XXXX, and Error).
X2		Report basic call progress result codes and connections speeds—that is, OK, Connect, Ring, No Carrier (also, for busy, if enabled, and dial tone not detected: No Answer, Connect XXXX, and Error).
X3		Report basic call progress result codes and connection rate—that is, OK, Connect, Ring, No Carrier, No Answer, Connect XXXX, Busy, and Error.

* Default

After a command has been entered at the command line, the modem attempts to execute the command and then returns a result code to the screen. Table 11.5 describes the command result codes.

Table 11.5

AT Command result codes.

Result	Code	Description
0	OK	The OK code is returned by the modem to acknowledge execution of a command line.
1	CONNECT	The modem sends this result code when line speed is 300 bps.
2	RING	The modem sends this result code when incoming ringing is detected on the line.
3	NO CARRIER	Carrier not detected within time limit, or carrier lost.
4	ERROR	The modem could not process the command line (entry error).
5	CONNECT 1200	The modem detected a carrier at 1,200 bps.
6	NO DIAL TONE	The modem could not detect a dial tone when dialing.
7	BUSY	Modem detected a busy signal.
8	NO ANSWER	Modern never detected silence (@ command only).
9	CONNECT 0600	The modem sends this result code when line speed is 600 bps.
10	CONNECT 2400	The modem detected a carrier at 2,400 bps.
11	CONNECT 4800	Connection established at 4,800 bps.
12	CONNECT 9600	Connection established at 9,600 bps result code when line speed is 7,200 bps.
13	CONNECT 7200	The modem sends this result code when line speed is 7,200 bps.
14	CONNECT 12000	Connection established at 12,000 bps.

continues

Table 11.5 Continued

Result	Code	Description
15	CONNECT 14400	Connection established at 14,400 bps.
17	CONNECT 38400	Connection established at 38,400 bps.
18	CONNECT 57600	Connection established at 57,600 bps.
22	CONNECT 75TX/1200RX	The modem sends this result code upon establishing a v.23 originate.
23	CONNECT 1200TX/175RX	The modem sends this result code upon establishing a v.23 answer.
24	DELAYED	The modem returns code when a call fails to connect and is considered delayed.
92	BLACKLISTED	The modem returns this result code when an f\call fails to connect and is considered "blacklisted."
40	CARRIER 300	Carrier detected at 300 bps.
44	CARRIER 1200/75	The modem sends this result code when v.23 backward channel carrier is detected.
45	CARRIER 75/1200	The modem sends this result code when v.23 forward channel carrier is detected.
46	CARRIER 1200	Carrier detected at 1,200 bps.
47	CARRIER 2400	Carrier detected at 2,400 bps.
48	CARRIER 4800	The modem sends this result code when either the high or low channel carrier in v.22bis modem has been detected.
49	CARRIER 7200	Carrier detected at 7,200 bps.
50	CARRIER 9600	Carrier detected at 9,600 bps.
51	CARRIER 12000	Carrier detected at 12,000 bps.
52	CARRIER 14400	Carrier detected at 14,400 bps.

Result	Code	Description
66	COMPRESSION: CLASS 5	MNP Class 5 is active CLASS 5.
67	COMPRESSION: V.42bis	v.42bis is active v.42bis.
69	COMPRESSION: NONE	No Data Compression NONE.
70	PROTOCOL: NONE	No error correction.
77	PROTOCOL: LAPM	v.42 LAP-M error correction.
80	PROTOCOL: ALT	MNP Class 4 error correction.

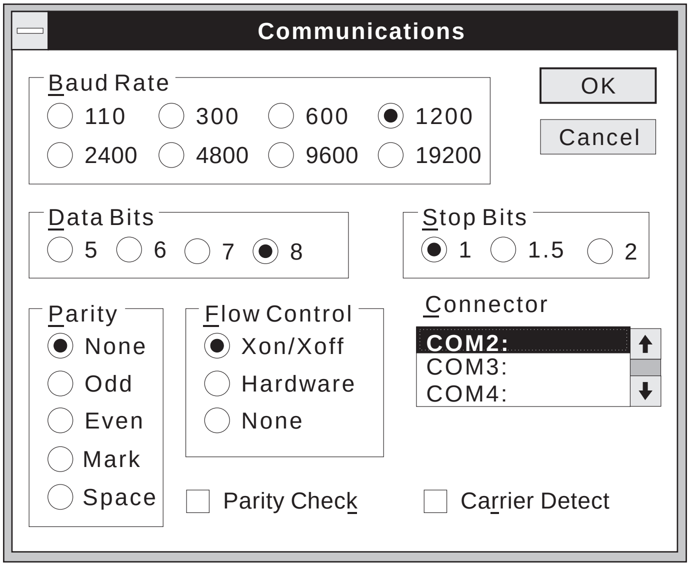
Specialized fax and voice software programs are also included, if the modem has these capabilities. Most communication software packages include a phone book to hold frequently called numbers. Other features include the use of a variety of different software protocols. Common protocols included at this level include Xmodem, Ymodem, and Zmodem. For the session to be successful, both the originating and answering modems must agree on the same protocol, baud rate, and data length.

To communicate with other computers, some information about how the communication will proceed is needed. In particular, it will be necessary to match the protocol of the remote unit, as well as its parity, character-framing, and baud-rate settings. With older modems, this may involve a telephone call to the other computer user. In the case of online services, the information is provided in the introductory package the user receives after the service is joined.

The Windows 3.x program contains an application, called `Terminal`, that controls the operation of the system's modem. Terminal is capable of operating with several different modem configurations. This flexibility allows it to conduct transfers with a wide variety of other computer systems. The Communications dialog box under the Settings option, shown in Figure 11.13, provides the options for configuring the communications settings.

Figure 11.13

The Communications setting dialog box.



Troubleshooting Modems

Objective

A section on troubleshooting modems has to be mentally divided into two segments:

- External modems
- Internal modems

In the case of an internal modem, it is checked out in the same basic sequence as any other I/O card. First, check the modem's hardware and software configuration, check the system for conflicts, and check for correct drivers. Improper software setup is the most common cause of modems not working when they are first installed. Inspect any cabling connections to see that they are made correctly and functioning properly, and test the modem's hardware by substitution. If an external modem is being checked, it must be treated as an external peripheral, with the serial port being treated as a separate I/O port. Figure 11.14 shows the components associated with internal and external modems.

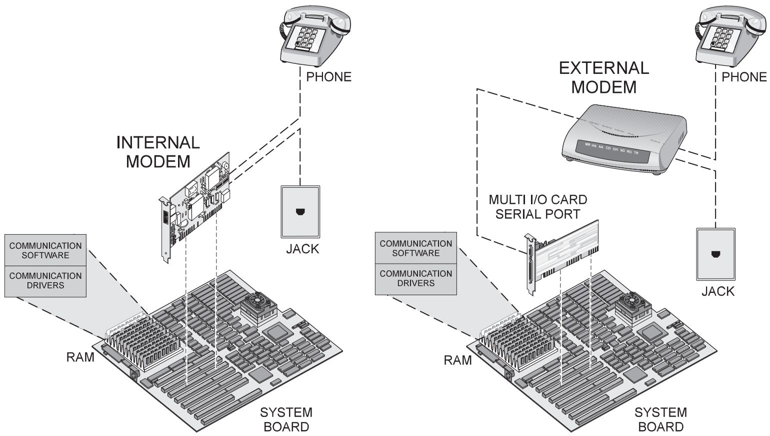


Figure 11.14

Internal and external modem components.

COM Port Conflicts

As stated earlier, every COM port on a PC requires an IRQ line to signal the processor for attention. In most PC systems, two COM ports share the same IRQ line. The IRQ4 line works for COM1 and COM3, and the IRQ3 line works for COM2 and COM4. This is common in PC compatibles. The technician must make sure that two devices are not set up to use the same IRQ channel. If more than one device is connected to the same IRQ line, a conflict will occur, because it is not likely that the interrupt handler software will be able to service both devices.

If a mouse is set for COM1 and a modem is set for COM3, for example, neither device will be able to communicate effectively with the system because COM1 and COM3 both use IRQ4. Both the mouse and the modem may be interrupting the microprocessor at the same time. The same is true if two devices are connected to IRQ3, because COM2 and COM4 use this IRQ. Therefore, the first step to take when installing a modem is to check the system to see how its interrupts and COM ports are allocated. This particular interrupt conflict may be alleviated by using a bus mouse rather than a serial mouse, thereby freeing up a COM port.

Use a software diagnostic package to obtain information about the serial port's `base I/O port address`. A typical value for this setting is `02E8`. Also, obtain the modem's Interrupt Request Line (IRQ) setting. This setting is typically `IRQ=3`. Other common modem settings are as follows:

COM1 with `IRQ=4, 5, or 7`

COM2 with `IRQ=3, 5, or 7`

COM3 with `IRQ=4, 5, or 7`

COM4 with `IRQ=3, 5, or 7`

Modem Software Checks

Many of the software diagnostic packages available include a utility for testing modems. If such a program is available, run the equivalent of its Run All Tests entry to test the modem. If all the configuration settings are correct, attempt to run the modem's DOS-based communications package to test the modem's operation. At the command line, type **ATZ** to enter the command mode using the Hayes-compatible command set. You should receive a 0, or OK response, if the command was processed.

If no result code is returned to the screen, check the modem's configuration and setup again for conflicts. Also, check the speed setting of the communication software to make sure it is compatible with that of the modem. On the other hand, a returned OK code indicates that the modem and the computer are communicating properly.

Other AT-compatible commands can be used to check the modem at the DOS level. The `ATL2` command sets the modem's output volume to medium to make sure that it is not set too low to be heard. If the modem dials, but cannot connect to a remote station, check the modem's speed and DTR settings. Change the DTR setting by entering **AT&D*n***. When:

n = 0 the modem ignores the DTR line.

n = 1 the modem goes to asynch command state when DTR goes off.

n = 2 DTR off switches modem to off-hook, and back to command mode.

n = 3 DTR switching off initializes modem.

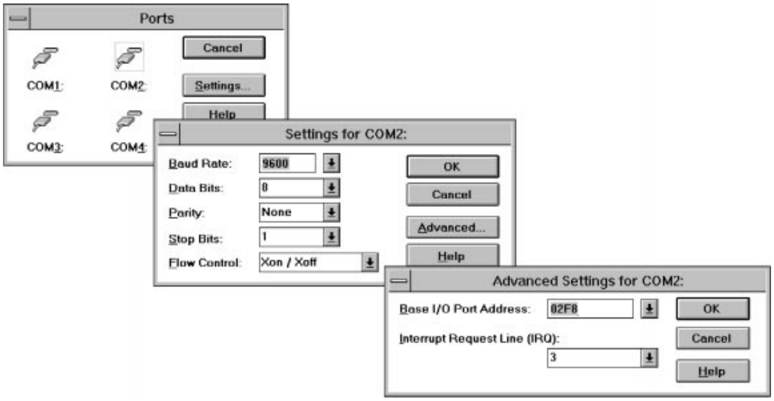
If the modem connects but cannot communicate, check the character-framing parameter of the receiving modem and set the local modem to match. Also, match the terminal emulation of the local unit to that of the remote unit. ANSI terminal emulation is the most common. Finally, match the file transfer protocol to the other modem.

Use the `ATDT70*` command to disable call waiting, if the transmission is frequently garbled. The +++ command should interrupt any activity the modem is engaged in and bring it to the command mode.

In Windows 3.x, the modem-related drivers are located in the Ports section of the Control Panel. The configuration information is found under the Settings and Advanced Settings buttons, as described in Figure 11.15.

Figure 11.15

Configuration information.



In Windows 95, the modem-configuration information is found in the Control Panel under the Modems icon. Under the icon are two tabs: the General tab and the Diagnostics tab. The Properties button in the General window provides Port and Maximum Speed settings. The Connection tab provides character-framing information, as illustrated in Figure 11.16. The Connection tab's Advanced button provides error and flow-control settings, as well as modulation type.

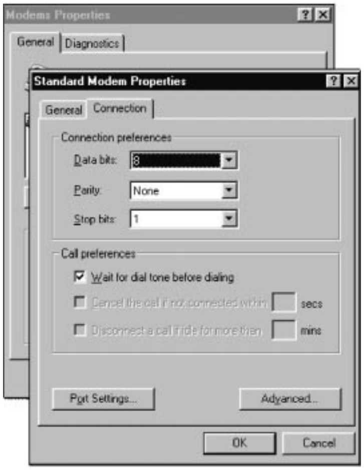


Figure 11.16

The Connection tab of the Standard Modem Properties dialog box.

The Diagnostics tab's dialog box, depicted in Figure 11.17, provides access to the modem's driver and additional information. The Plug and Play feature reads the modem card and returns its information to the screen, as demonstrated in the depiction.

Windows 95 provides fundamental troubleshooting information for wide area networking through its system of Help screens. Just select Help from the Control Panel's toolbar, and then click on the topic that you are troubleshooting.

Modem Hardware Checks

From the earlier section concerning UART registers, it should be apparent that modems contain the capability to perform self-tests on sections of their circuitry. Modems actually have the capability to perform three different kinds of self-diagnostic possessive tests. The first is a `local digital loopback test`, where data is looped through the UART's internal registers. When testing the RS-232 port itself, a device called a `wrap-plug` channels the output data directly back into the received data input, and only the port is tested. Many modems have the capability to extend this test by looping the data through the local modem and back to the computer (`local analog loopback test`). Some modems even possess the capability to loop-back data to a remote computer through its modem (`remote digital loopback test`). In this manner, the entire transmit and receive path can be validated, including the communication line (that is, the telephone line). One of the most overlooked causes of transmission problems is the telephone line itself. A noisy line can easily cause garbled data to be output from the modem. Figure 11.18 illustrates adapter, analog, and digital loopback tests.

Figure 11.17

The Diagnostics tab of the Modems Properties dialog box.

If the modem is an internal unit, its hardware can be tested by exchanging it with a known good unit. If the telephone line operates correctly with a normal handset, only the modem card, its configuration, or the communications software can be causes for problems. If the software and configuration settings appear correct, it will be necessary to exchange the modem card.

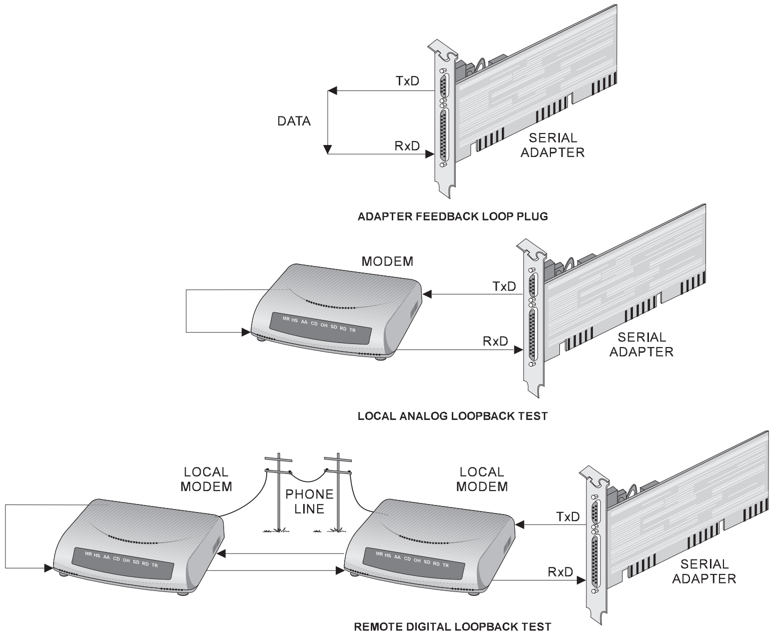


Figure 11.18

Loopback tests.

With an external modem, the front-panel lights can be used as diagnostic tools to monitor its operation. The progress of a call, and its handling, can be monitored along with any errors that may occur. Figure 11.19 depicts the front-panel lights of a typical external modem.

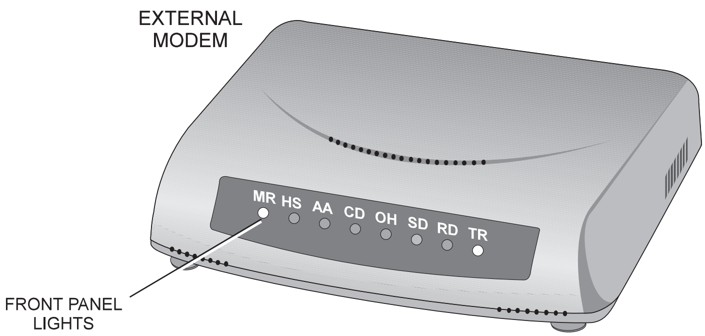


Figure 11.19

Modem front panel.

The modem ready (MR), terminal ready (TR), and autoanswer (AA) lights are preparatory lights that indicate that the modem is plugged in, powered on, ready to run, and prepared to answer an incoming call. The MR light becomes active when power is applied to the modem and it is ready to operate. The TR light becomes active when the host computer's communication software and the modem contact each other. The AA light just indicates that the autoanswer function has been turned on.

The off-hook (OH), ring indicator (RI), and carrier detect (CD) lights indicate the modem's online condition. The off-hook light indicates that the modem has connected to the phone line. This action can occur when the modem is receiving a call or when it is commanded to place a call. The RI light becomes active when the modem detects an incoming ring signal. The CD light becomes active when the modem detects a carrier signal from a remote modem. As long as this light is on, the modem is capable of sending and receiving data from the remote unit. If the CD light will not become active with a known good modem, a problem with the data communication equipment (DCE) exists.

The final trio of lights indicate the status of a call in progress. The send data (SD) light flickers when the modem transmits data to the remote unit; the received data light flickers when the modem receives data from the remote unit. The high speed (HS) light becomes active when the modem is conducting transfers at its highest-possible rate. If an external modem will not operate at its highest-rated potential, the specification for the UART on the adapter card should be checked to make certain that it is capable of operating at that speed.

Local Area Networks

Objective

Local area networks (LANs) are systems designed to connect computers together in a relatively close proximity. These connections enable users attached to the network to share resources such as printers and modems. LAN connections also enable users to communicate with each other and to share data among their computers.

When discussing LANs, there are two basic topics to consider: the LAN's `topology` (hardware connection method) and its protocol (communication control method). In concept, a minimum of three stations must be connected to have a true LAN. If only two units are connected, point-to-point communications software and a simple null modem could be used.

LAN Topologies

Network topologies are connection/configuration strategies. LAN topologies fall into the following three types of configurations:

- Bus
- Ring
- Star

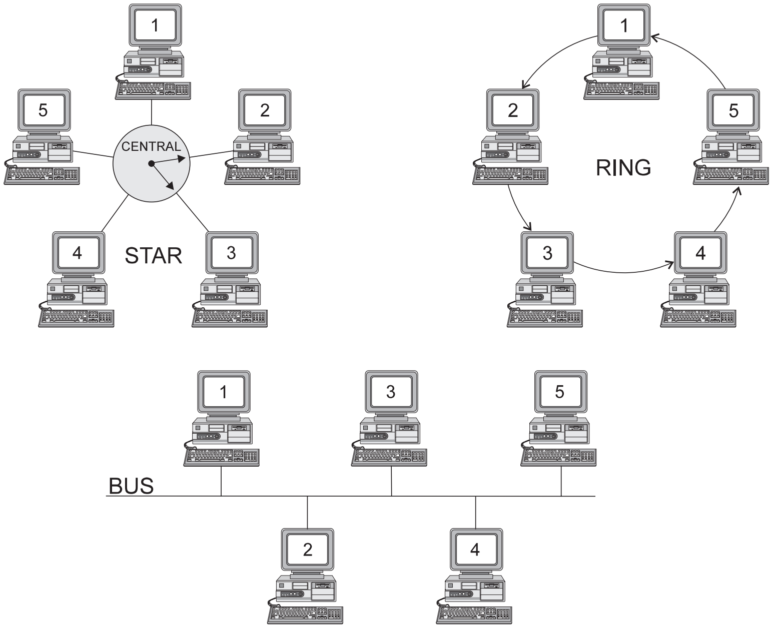
All three topologies are illustrated in Figure 11.20. In the bus topology, the stations, or `nodes`, of the network connect to a central communication link. Each node has a unique address, along the bus, that differentiates it from the other users on the network. Information can be placed on the bus by any node. The information must contain network address information about the node(s) that the information is intended for. Other nodes along the bus will ignore the information.

In a ring network configuration, the communication bus is formed into a closed loop. Each node inspects the information on the LAN as it passes by. A `repeater`, built in to the ring LAN card, regenerates every message not directed to it and sends it to the next node. The originating node eventually receives the message back, and removes it from the ring.

In a star topology, the logical layout of the network resembles the branches of a tree. All the nodes are connected, in `branches`, that eventually lead back to a central unit. Nodes communicate with each other through the central unit. The central station coordinates the network's activity by polling the nodes, one by one, to see whether they have any information to transfer. If so, the central station gives that node a predetermined slice of time to transmit. If the message is longer than the time allotted, the transmissions is chopped into small segments that are transmitted over several polling cycles.

Figure 11.20

Star, bus, and ring configurations.



Control of the network can be implemented in two ways: as `peer-to-peer networks`, where each computer is attached to the network in a ring or bus fashion; and `client/server networks`, where workstations (referred to as clients) operate in conjunction with a dedicated, master (`file server`) computer.

Consider the typical peer-to-peer network arrangement. In this arrangement, the users connected to the network can share access to different network resources such as one another's hard drives and printers. However, control of the local unit is fairly autonomous. The nodes in this type of network configuration usually contain local hard drives and printers that the local computer has control of. The items can be shared at the discretion of the individual user.

A typical client/server LAN configuration is depicted in Figure 11.21. In this type of LAN, control tends to be very centralized. The server(s) typically holds the programs and data for all the LAN's users. In some cases, the client units do not even include a local hard or floppy-disk drive unit. Bootup is performed through an onboard BIOS, and no data is ever stored in the client machine. This type of client is referred to as a `workstation`.

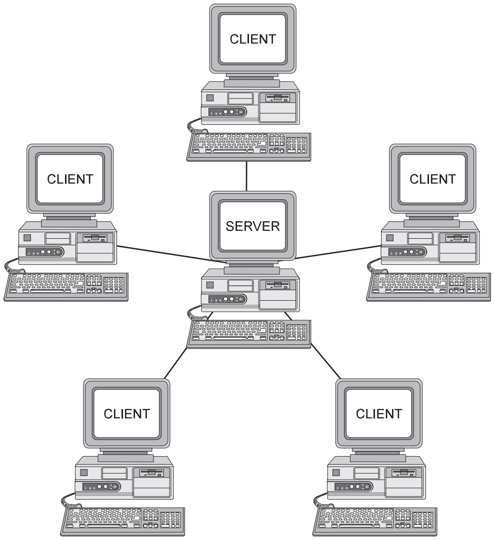


Figure 11.21

A client/server network.

Network Protocols

In a network, some method must be used to determine which node has use of the network's communications paths, and for how long it can have it. The network's protocol handles these functions, and is necessary to prevent more than one user from accessing the bus at any given time. If two sets of data are placed on the network at the same time, a `data collision` occurs, and data is lost. Basically, there have been three de facto networking protocols. These are `Ethernet`, `ARCnet`, and `Token Ring`.

Ethernet

Objective

Ethernet was developed by Xerox in 1976. Its methodology for control is referred to as `Carrier Sense Multiple Access with Collision Detection` (CSMA/CD). Using this protocol, a node that wants to transfer data over the network first listens to the LAN to see whether it is in use. If not, the node begins transmitting its data. If the network is busy, the node waits for the LAN to clear for a predetermined time, and then takes control of the LAN.

If two nodes are waiting to use the LAN, they will periodically attempt to access the LAN at the same time. When this happens, a data collision occurs and the data from both nodes is rendered useless. The receiver portion of the Ethernet controller monitors the transmission to detect collisions. When it senses the data bits overlapping, it halts the transmission, as does the other node. The transmitting controller generates an `Abort Pattern Code` that is transmitted to all the nodes on the LAN, telling them that a collision has occurred. This alerts any nodes that might be waiting to access the LAN that there is a problem.

The receiving node (or nodes) dumps any data that it might have received before the collision occurred. Other nodes waiting to send data generate a random timing number and go into a holding pattern. The timing number is a waiting time that the node sits out before it tries to transmit. Because the number is randomly generated, the odds against two of the nodes trying to transmit again at the same time is very low. The first node to timeout listens to the LAN to see whether any activity is still occurring. Because it almost always finds a clear LAN, it begins transmitting. If two of the nodes do timeout at the same time, another collision happens and the abort pattern/number generation/timeout sequence begins again. Eventually, one of the nodes will gain clear access to the network and successfully transmit its data.

The Ethernet strategy allows for up to 1,024 users to share the LAN. From the description of its collision-recovery technique, however, it should be clear that with more users on an Ethernet LAN, more collisions are likely to occur, and the average time to complete an actual data transfer will be longer.

The Ethernet Frame

Under the Ethernet standard, information is collected into a package called a `frame`. A typical Ethernet frame is depicted in

Figure 11.22. The frame carries the following six sections of information:

- A preamble
- A destination address
- An originating address
- A type field
- The data field
- The frame check error-detection and correction information

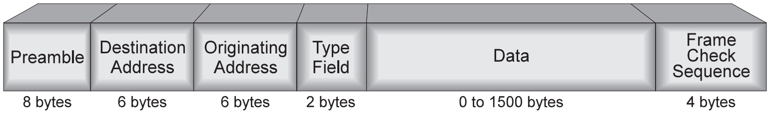


Figure 11.22

A typical Ethernet frame.

This organizational structure is very similar to that of a sector on a hard disk. The preamble synchronizes the receiver's operation to that of the transmitter. This action also tells the other nodes that a transmission is under way. The Ethernet preamble is a 64-bit string, made up of alternating 1's and 0's, that ends in two consecutive 1's.

The destination address field is 6 bytes long and is used to define one of three address locations. This number can represent the individual node address of the intended receiver, the address of a grouping of nodes around the LAN, or it can be a broadcast code that allows the node to send a message to everyone on the LAN.

The originating address field contains the identification address for the transmitting node. The type field is a 2-byte field that identifies the user protocol of the frame.

The data field is a variable-length field that contains the actual information. Because it is sent in asynchronous mode, the data field can be as long as necessary. However, the Ethernet standard does not allow for data fields less than 46 bytes, or longer than 1,500 bytes.

The frame-check block contains an error-detection and correction word. Like parity and other error-detection schemes, the receiver regenerates the error code from the received data (actually the data, the address bytes, and the type field) and compares it to the received code. If a mismatch occurs, an error signal is generated from the LAN card to the system.

Ethernet Specifications

Ethernet is classified as a bus topology. The original Ethernet scheme was classified as a 10MHz transmission protocol. The maximum length specified for Ethernet is 1.55 miles (2.5 km), with a maximum segment length between nodes of 500 meters. This type of LAN is referred to as a `10base5` LAN by the IEEE organization. However, newer implementations, called `Fast Ethernet`, are producing LAN speeds of up to 100 Mbps.

Ethernet connections can be made through 50-ohm, coaxial cable (10base5), thinnet coaxial cable (10base2), or unshielded twisted-pair (UTP) cabling (1base5). The Fast Ethernet specifications that use UTP are referred to as 10BaseT (10Mbps) and 100BaseT (100Mbps). Ethernet cards capable of supporting these transmission rates are classified as 10/100 cards.

Coaxial cable is familiar to most people as the conductor that carries cable TV into their homes. Coax has a single copper conductor in its center and a protective braided-copper shield around it. UTP cable is common telephone cabling with four pairs of twisted wires inside.

The 10base2 Ethernet LAN uses thinner, industry-standard RG-58 coaxial cable, and has a maximum segment length of 185 meters. The UTP specification is based on telephone cable and is normally used to connect a small number of PCs together. The twisted pairing of the cables uses magnetic-field principles to minimize induced noise in the lines. UTP has a transmission rate that is stated as 1Mbps. Using UTP cable, a LAN containing up to 64 nodes can be constructed with the maximum distance between nodes set at 250 meters. Typical coaxial and UTP connections are depicted in Figure 11.23.

The original 10base5 connection scheme required that special `transceiver units` be clamped to the cable. A pin in the transceiver pierced the cable to establish electrical contact with its conductor. An additional length of cabling, called the `drop cable`, was then connected between the LAN adapter card and the transceiver.

In a 10base2 LAN, the node's LAN adapter card is usually connected directly to the LAN cabling, using a T-connector (for peer-to-peer networks), or by a BNC connector in a client/server LAN. The 1base5 system uses RJ-45 jacks and plugs to connect the LAN together. RJ-45 connectors are very similar in appearance to the RJ-11 connectors used with telephones and modems. However, the RJ-45 connectors are considerably larger than the RJ-11 connectors. Some Ethernet adapters include 15-pin sockets that allow special systems, such as fiber-optic cable, to be interfaced to them. Other cards provide specialized ST connectors for fiber-optic connections.

1base5 systems normally use `concentrators`, like the one in Figure 11.24, for connection purposes. Both coaxial connection methods require that a terminating resistor be installed at each end of the transmission line. Ethernet systems use 52-ohm terminators.

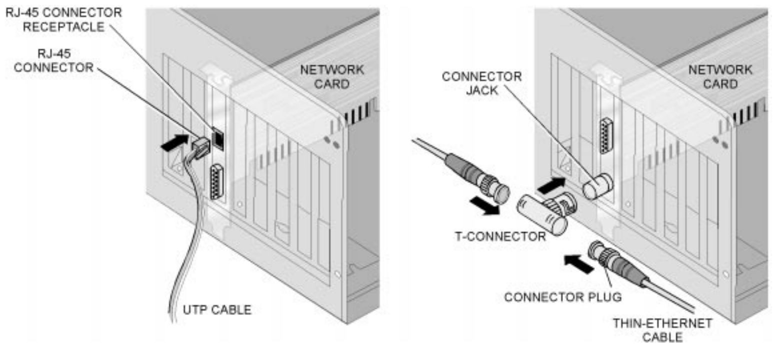


Figure 11.23

Typical coax and UTP connections.

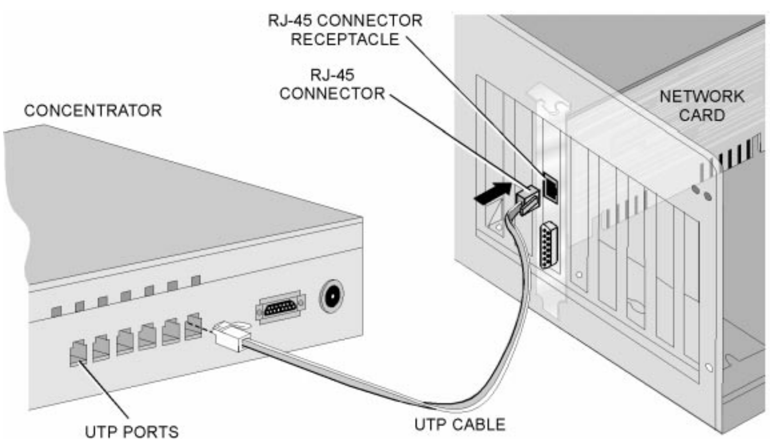


Figure 11.24

UTP between a computer and a concentrator.

Fiber-optic cable offers the prospects of very high performance links for LAN implementation. It can handle much higher data transfer rates than copper conductors and can use longer distances between stations before signal deterioration becomes a problem. In addition, fiber-optic cable offers a high degree of security for data communications—because it does not radiate EMI signal information that can be detected outside the conductor, it does not tap easily, and it shows a decided signal loss when it is tapped into.

Although fiber-optic cable is now widely used in telephone systems, and there are some standards in place for implementing LANs on fiber-optic cabling, few LAN hardware manufacturers have switched over to fiber-optic networking at this time.

Table 11.6 summarizes the different Ethernet specifications. Other CSMA/CD-based protocols exist in the market. Some are actually Ethernet compatible. These systems may or may not, however, achieve the performance levels of a true Ethernet system. Some may actually perform better.

Table 11.6

Ethernet specifications.

Classification	Conductor	Maximum Segment	Nodes Length	Maximum Rate	Transfer Length
1Base5	UTP	250 m	64	500 m	1Mbps
10Base2	RG-58	185 m	30	250 m	10Mbps
10Base5	RG-50	500 m	10	2.5 km	10Mbps

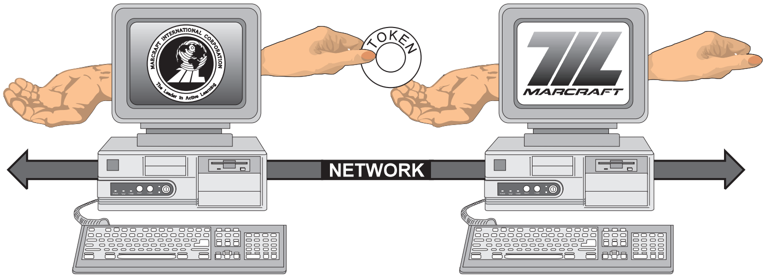
ARCnet

Objective

The `ARCnet` (Attached Resource Computer Network) protocol was developed by Datapoint, and is based on a modified `token-passing` scheme. In a token-passing system, contention for use of the LAN between different nodes is handled by passing an electronic enabling code, called a `token`, from node to node. Only the node possessing the token can have control of the LAN to transmit information. This concept is illustrated in Figure 11.25.

Figure 11.25

A token-passing scheme.



The token is passed from node to node along the LAN. Each node is allowed to hold the token a prescribed amount of time. After sending its message, or after its time runs out, the node must transfer the token to the next node. If the next node has no message, it just passes the token along to the next designated node. Nodes do not have to be in numerical sequence. (Their sequences are programmed in the network management software.) All nodes listen to the LAN during the token-passing time.

In a token-passing network, new or removed nodes must be added to, or deleted from, the rotational list in the network-management software. If not, the LAN will never grant access to the new nodes. Most ARCnet management software and cards are built so that each device attached to the LAN is interrogated when the LAN is started up. In this way, the rotational file is verified each time the network is started.

New nodes that start up after the LAN has been initialized transmit a reconfiguration burst that can be heard by all the nodes. This burst grabs the attention of all the installed nodes and erases their token destination addresses. Each node goes into a wait state that is determined by its station number. The node with the highest station number times out first and tries to access the LAN.

The highest-numbered node is responsible for starting the token-passing action after a new unit is added to the LAN. This is accomplished by broadcasting a signal to all nodes telling them what it believes is the lowest-numbered node in the LAN, and asking whether it will accept the token. If no response is given, the node moves to the next known address in the LAN management-software's roster and repeats the request. This action continues until an enabled node responds. At this point, the token is passed to the new node and the forwarding address is stored in the LAN manager's list. Each successive node goes through the same process until all the nodes have been accessed.

The node passing the token must always monitor the LAN, under the ARCnet scenario. This is done to prevent the loss of the token during its passage. If the node remains inactive for a predetermined amount of time, the transmitting node must reclaim the token and search for the next active node to pass it to. In this case, the transmitting node just increments its next-node address by one, and attempts to make contact with a node at that address. If not, it will increment the count by one again and retry until it reaches an enabled node. This new node number is stored in the transmitting node, and will be the pass-to number for that node until the system is shut down or reconfigured.

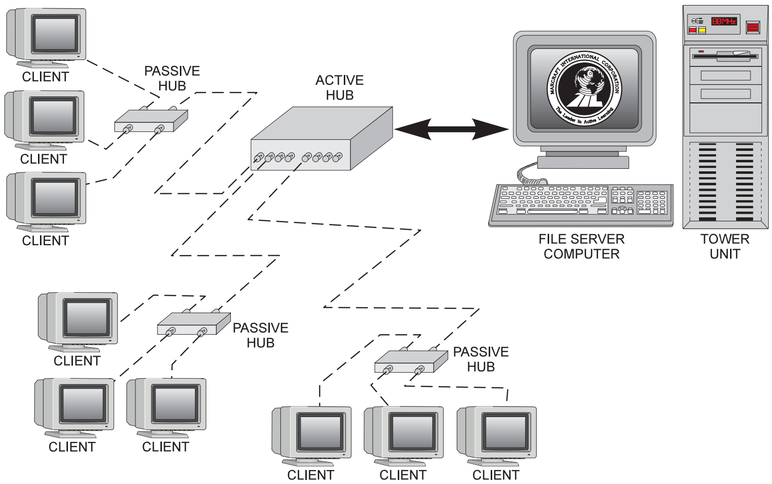
ARCnet Specifications

In the ARCnet specification, no particular transmission medium is defined. ARCnet can be used with coax cable, twisted-pair cable, or fiber-optic cable. ARCnet coaxial cabling is defined as RG-59 cable, with 75-ohm terminators required for any open nodes. The ARCnet protocol transfers data at a nominal rate of 2.5 MHz. As with the Ethernet rating, this is a maximum rate. Obviously, neither network can push through data at this rate if it is being transmitted to a 1200-bps modem or a serial printer. Other system items, such as hard drives and video adapters, can also limit the true speed of the network.

Although the ARCnet topology is defined as a ring, its actual construction resembles more of a star or tree structure, as illustrated in Figure 11.26. Each station connects to a signal repeater, called a `hub`. Hubs may be active or passive. `Active hubs` actually receive the signal, regenerate it, and resend it. The maximum distance a node can be located from an active hub is 600 meters; the maximum distance from a `passive hub` is 30 meters. Hubs can be linked together to increase the actual distance between nodes.

Figure 11.26

The ARCnet tree structure.

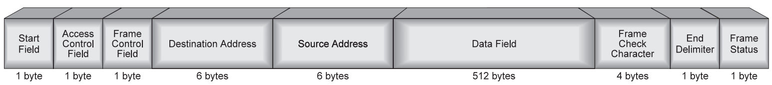


ARCnet Frames

The packet construction for an ARCnet frame is depicted in Figure 11.27. It uses a 1-byte Start field, a 1-byte Access Control field, a 1-byte Frame Control field, 6 bytes of destination address, 6 bytes of source address, a 512-byte (maximum) Data field, a 4-byte Frame Check character, a 1-byte End Delimiter mark, and a 1-byte Frame Status field.

Figure 11.27

Packet construction for an ARCnet frame.



The Start field contains bits that the nodes can detect as markers for the start of a transmission. The Access Control field establishes the token priority system. One bit is a `monitor bit` that is used by the transmitting node to identify a frame that it should remove from the LAN. The Frame Control byte declares which type of frame is being transmitted (that is, a ring management frame or a data frame). The Source and Destination frames should be self-explanatory, as should the Data frame.

The Frame Check field is a 32-bit checksum character that detects single-bit transmission errors. The End Delimiter is an encoded bit-pattern that tells the receiving node that the frame is ending. One of its bits is used in multipacket transmissions to indicate whether the current frame is an intermediate or final frame. Finally, the Frame Status field is used to determine whether the receiving node is present on the LAN and whether it has received the frame. Because the receiver actually copies the frame and places it back on the LAN, it has an opportunity to set the received bit in this field if it has been received. If not, the bit remains unchanged and will be recognized as an unreceived frame when it gets back to the original node.

Token Ring

In 1985, IBM developed a token-passing LAN protocol it called the Token Ring. As its name implies, Token Ring is a token-passing protocol operating on a ring topology. The token is a small frame that all nodes can recognize instantly. When the ring is idle, the token just passes counterclockwise from station to station. A station can transmit information on the ring any time it receives the token. This is accomplished by turning the token packet into the start of a data packet, and adding its data to it.

At the intended receiver, the packet is copied into a buffer memory, serially, where it is stored. The receiver also places the packet back on the ring so that the transmitter can get it back. Upon doing so, the transmitter reconstructs the token and places it back on the ring. This concept is described in Figure 11.28.

IBM uses `trunk coupling units` (TCUs) to connect to the cable, as described in Figure 11.29. The cabling is a two-pair, shielded twisted-pair cable. The main cable is called the `trunk cable`, and the individual drops are referred to as the `interface cable`. The TCUs are grouped together in units called concentrators. Internally, the concentrator's ports are connected into a ring configuration. In this manner, the concentrator can be placed in a convenient area and have nodes positioned where they are needed. Some Token Ring adapters provide 9-pin connectors for `Shielded Twisted Pair` (STP) cables as well.

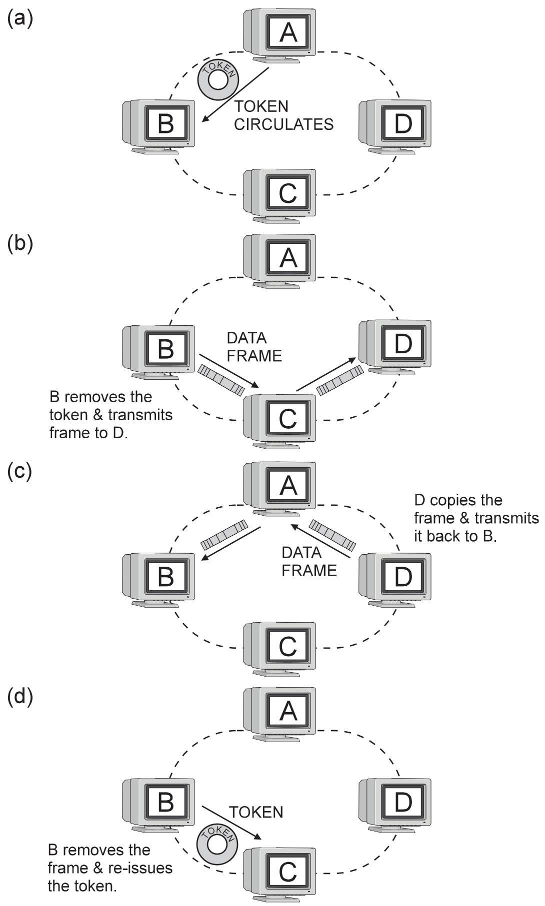


Figure 11.28

Token-Ring concept.

The frame packaging for a Token-Ring transmission is very similar to that of an ARCnet frame. The data transfer rate stated for Token-Ring systems is 4 to 16 Mbps. Token-passing is less efficient than other protocols when the load on the network is light. However, it evenly divides the network's usage between nodes when traffic is heavy. It can also be extremely vulnerable to node crashes when that node has the token. LAN adapter cards are typically designed to monitor the LAN for such occurrences so that they can be corrected without shutting down the entire network.

Table 11.7 lists the specifications of the ARCnet and Token-Ring topologies.

Figure 11.29

TCU connections.

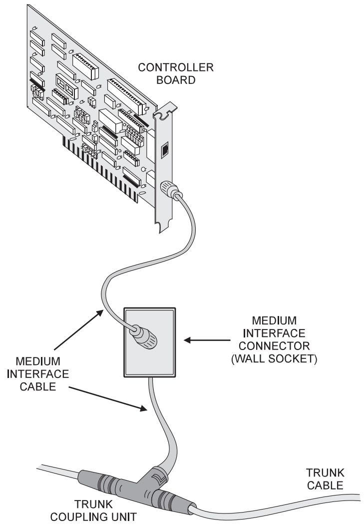


Table 11.7

Summary of ARCnet and T/Ring specs.

Classification	Conductor	Maximum Segment Length	Nodes	Transfer Rate
ARCnet	RG-59(75*)	600 m	9/17	2.5Mbps
Token Ring	STP(150*)	185 m	30	4Mbps

LAN Adapter Cards

In a LAN, each computer on the network requires a network adapter card (also referred to as a NIC), and every unit is connected to the network by some type of cabling. These cables are typically either twisted-pair wires, thick or thin coaxial cable, or fiber-optic cable. When dealing with a LAN adapter card, it must have connectors compatible with the type of LAN cabling being used. Many Ethernet LAN cards come with both an RJ-45 and a BNC connector so that the card can be used in any type of Ethernet configuration.

A typical LAN card is depicted in Figure 11.30. In addition to its LAN connectors, the LAN card may have a number of configuration jumpers that must be set up. Although some cards may have jumper instructions printed directly on the I/O card, the card's User's Manual is normally a must for configuring the card for operation. Great care should be taken with the User's Manual because its loss might render the card useless. At the very least, the manufacturer would have to be contacted to get a replacement.

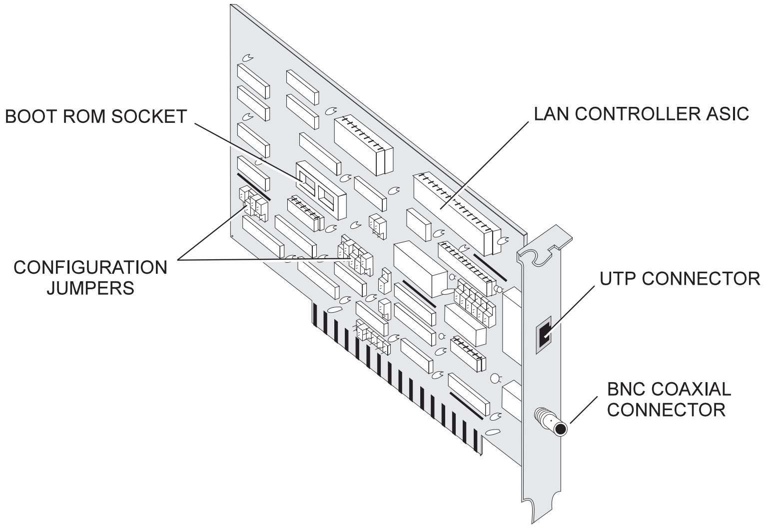


Figure 11.30

A typical LAN card.

Another item of note on LAN cards includes the presence of a vacant ROM socket. This socket can be used to install a `bootup ROM` to allow the unit to be used as a `diskless workstation`. One or more activity lights may also be included on the card's back plate. These lights can play a very important part in diagnosing problems with the LAN connection. Check the card's User's Manual for definitions of its activity lights.

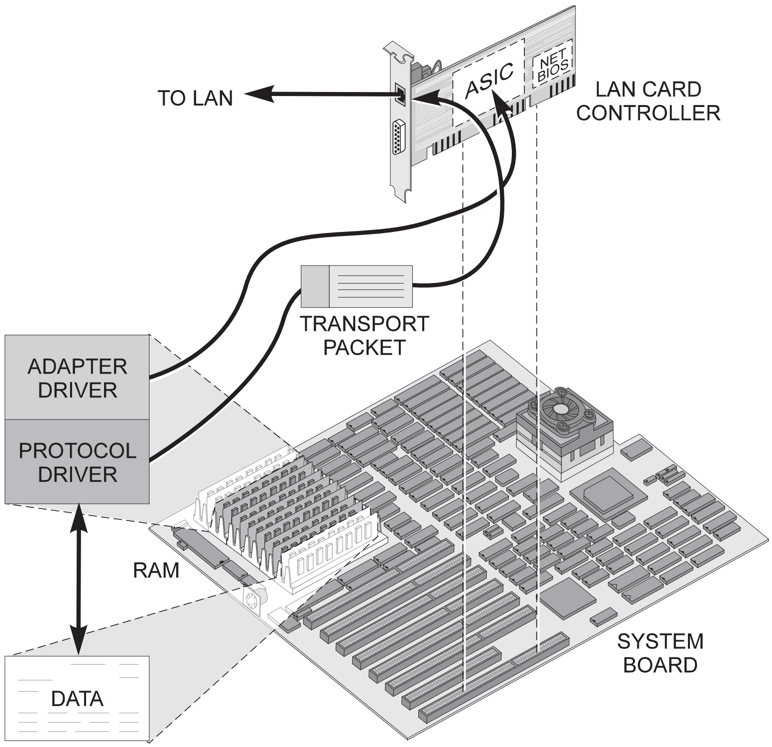
Each adapter must have an adapter driver program loaded in its host computer to handle communications between the system and the adapter. These are the Ethernet, ARCnet, and Token-Ring drivers loaded to control specific types of LAN adapter cards.

In addition to the adapter drivers, the network computer must have a network protocol driver loaded. This program may be referred to as the `transport protocol`, or just as the protocol. It operates between the adapter and the initial layer of network software to package and unpackage data for the LAN. In many cases, the computer may have several different protocol drivers loaded so that the unit can communicate with computers that use other types of protocols.

Typical protocol drivers include the Internetworking Packet Exchange/Sequential Packet Exchange (IPX/SPX) model produced by Novell and the standard Transmission Control Protocol/Internet Protocol (TCP/IP) developed by the U.S. military for its ARPA network. Figure 11.31 illustrates the various LAN drivers necessary to transmit or receive data on a network. More specific protocol information is provided throughout the remainder of this chapter.

Figure 11.31

Various LAN drivers.



Installing LANs

Installing a LAN card in a PC follows the basic steps of installing most peripheral cards. Check the system for currently installed drivers and system settings. Consult the LAN card's Installation Guide for default settings information and compare them to those of devices already installed in the system. If there are no apparent conflicts between the default settings and those already in use by the system, place the adapter card in a vacant expansion slot and secure it to the system unit's back plate.

Connect the LAN card to the network as directed by the manufacturer's Installation Guide and load the proper software drivers for the installed adapter (see Table 11.8). Figure 11.32 illustrates connecting the computer to the LAN, using UTP or coaxial cable. If UTP is being used, the line drop to the computer would come from a concentrator like the one depicted.

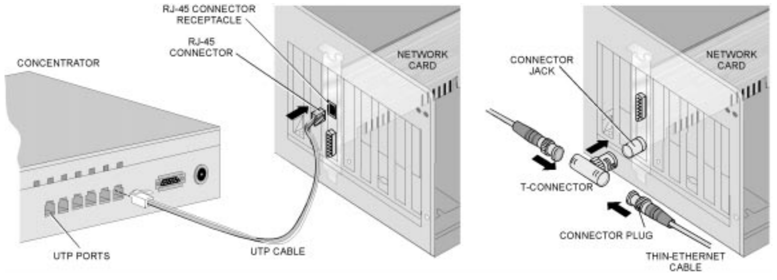


Figure 11.32

Connecting the computer to the LAN.

The following three pieces of important information are required to configure the LAN adapter card for use:

- The interrupt request (IRQ) setting the adapter will use to communicate with the system.
- The base I/O port address the adapter will use to exchange information with the system.
- The base memory address that the adapter will use as a starting point in memory for DMA transfers.

Some adapters may require that a DMA channel be defined for it.

Typical configuration settings for the network card's IRQ, I/O port address, and base memory are as follows:

IRQ=5

Port Address=300h

Base Memory=D8000h

If a configuration conflict appears, reset the conflicting settings so that they do not share the same value. Which component's configuration gets changed is determined by examining the options for changing the cards involved in the conflict. A sound card may have many more IRQ options available than a given network card. In this case, it would be easier to change sound card settings than network card settings.

Table 11.8

LAN card configuration settings.

I/O Address Options	Interrupt Request Channels	Extended Memory Addressing
240h	IRQ2	C000h
280h	IRQ3	C400h
2C0h	IRQ4	C800h
320h	IRQ10	CC00h
340h	IRQ11	D000h
360h	IRQ12	D400h
	IRQ15	DC00h

Troubleshooting LANs

Objective

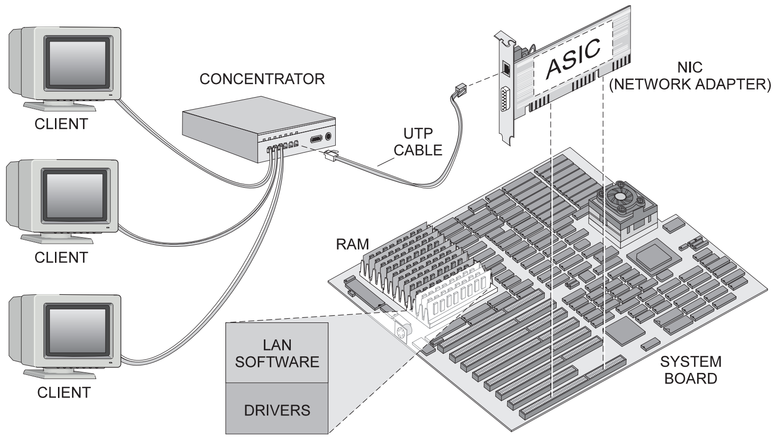
Begin troubleshooting a general network problem by determining what has changed since it was running last. If the installation is new, it will need to be inspected as a setup problem. Check to see whether any new hardware or new software has been added. Has any of the cabling been changed? Have any new protocols been added? Has any network adapter been replaced or moved? If any of these events has occurred, begin by checking them specifically.

If the system has not been changed, and has operated correctly in the past, the next step is to make certain that it functions properly as a standalone unit. Begin by disconnecting the unit from the network and testing its operation. Run diagnostics on the system to see whether any problems show up. If a hardware problem is encountered at the standalone level, troubleshoot the indicated portion of the system by using the procedures already discussed.

If the problem does not appear in, or is not related to, the standalone operation of the unit, it will be necessary to check the portions of the system that are specific to the network. These elements include the network adapter card, the network-specific portions of the operating system, and the network drop cabling. Figure 11.33 depicts the network-specific portions of a computer system.

Figure 11.33

Network-related components.



Be aware that in a network environment no unit really functions alone. Unlike working on a standalone unit, the steps performed on a network computer may affect the operation of other units on the network. Disconnecting a unit from a network that uses coaxial cable, for example, creates an unterminated condition in the network. This condition can cause several different types of problems, including the following:

- Data moving through the network can be lost.
- A general slowdown of data movement across the network can occur because of reduced bandwidth.
- Nodes may not be able to “see” or to connect to each other.

If a unit must be removed from the network, it is a good practice to place a terminator in the empty connector where the unit was attached. This should allow the other units to function without the problems associated with an open connection. Care must be taken to ensure that the proper value of the terminating resistor is used. Substituting a terminator from an ARCnet system into an Ethernet system may create as many problems as the open connection would have. However, they may be harder to track down. Systems that use concentrators have fewer connection problems when a unit needs to be removed for servicing.

Even if the unit does not need to be removed from the network, diagnostic efforts and tests run across the network can use up a lot of the network’s bandwidth. This reduced bandwidth causes the operation of all the units on the network to slow down. This results because of the added usage of the network.

Because performing work on the network can affect so many users, it is good practice to involve the network administrator in any such work being performed. This person can run interference for any work that must be performed that could disable the network or cause users to lose data.

LAN Configuration Checks

As with any peripheral device, its configuration must be correct for the software driving the peripheral and for the adapter card it is communicating through. An improperly configured network adapter card can prevent the system from gaining access to the network. Many newer network cards possess Plug-and-Play capabilities. With other network cards, such as ISA cards, it is necessary to configure the card through hardware jumpers, or through EPROM Configuration Switches. Check the adapter card’s hardware settings to see whether they are set according to the manufacturer’s default settings, or whether they have been changed to some new setting.

If they have been changed, refer to the system information from the software diagnostic tool to see whether there is some good explanation for the change. If not, record the settings as they are and reset them to their default values. Also, check the software's configuration settings and change them to match as necessary. If a defective card is being replaced with an identical unit, transfer the configuration setting to the new card.

Use a software diagnostic package, such as MSD, to check the system's interrupt request allocations. Try to use a package that has the capabilities to check the system's I/O port addresses and shadow RAM and ROM allocations. Finally, check the physical IRQ settings of any other adapter cards in the system.

LAN Software Checks

It is at the LAN system-software level that troubleshooting activities diverge. The differences between Novell's NetWare, Microsoft's Windows NT, Windows for Workgroups (Windows 3.11), and Windows 95 are significant enough that there are nationally recognized certifications just for NetWare and NT. Novell NetWare and Windows NT are client/server types of network-management software; Windows 3.11 and Windows 95 are peer-to-peer networking environments.

One of the major concerns in most network environments is data security. Because all the data around the network is potentially available to anyone else who attaches to the network, all LAN administration software employs different levels of security. Passwords are typically used at all software levels to lock people out of hardware systems as well as out of programs and data files.

Logon passwords and scripts are designed to keep personnel from accessing the system or its contents. Additional passwording may be used to provide access to some parts of the system but not others. (That is, lower-level accounting personnel may be allowed access to accounts receivable and payable sections of the business-management software package, but not allowed into the payroll section.) A series of passwords may be used to deny access to this area.

In other LAN management packages, access and privileges to programs and data can be established by the network administrator through the software's security system. These settings can be established to completely deny access to certain information or to allow limited access rights to it. An example of limited rights is the ability to read data from a file, but not be able to manipulate it (write, delete, print, or move it) in any way.

The reason for discussing security at this point is because established security settings can prevent the technician from using any, or all, of the system's resources. In addition, having limited access to programs can give them the appearance of being defective. Because of this, the service technician must work with the network administrator when checking a networked machine. The administrator can provide the access and the security relief needed to repair the system. The administrator can also keep you away from data that may not be any of your business.

Windows 3.11

In Windows 3.11, several network-related files and file segments are added to the Windows operating environment to give it peer-to-peer networking capabilities. If a problem exists in the networking operation of Windows 3.11, exit to the DOS level and check the CONFIG.SYS file for a line that reads `DEVICE=IFSHLP.SYS`. Also check the AUTOEXEC.BAT file for a line that reads `C:\WINDOWS\NET_START`.

In Windows 3.11, use the Windows Write program to check the SYSTEM.INI file for a new section titled `[network]`, and to check for the presence of the new NETWORK.INI file.

Check the network selection in Windows Setup for the proper driver. The Change Network Settings option provides access to three important buttons:

- The Networks button
- The Sharing button
- The Drivers button

The Networks button allows network support to be turned on and off. It also allows additional networks to be added to unit. The Sharing button is used to allow sharing of local files and printers with other nodes on the network. The Drivers button is used to install and remove LAN adapters and protocols. Windows 3.11 offers a long list of adapter drivers to select from, and has some capabilities to autodetect the type of adapter card installed in the unit. Several protocol programs are included that will allow Windows 3.11 stations to exchange information with other types of Microsoft networks.

If a compatible driver does not exist in Windows, you will need to place a manufacturer-supplied driver disk in the floppy-disk drive and click on the Have Disk entry to upload the OEM drivers and protocols needed to operate the card.

Windows 95

In Windows 95, the computer's networking information can be checked through the Network icon, under the Start/Settings/ Control Panel path. The network drivers are located under the Configuration tab. Double-click on the installed adapter card's driver to obtain its configuration information.

Windows 95 includes a network diagnostic tool, called `Net Diag`, that can be used to troubleshoot connectivity problems from a second (functional Windows 95) node on the network. To use the Net Diag program, start it from the command line of a second computer on the network. This establishes the second unit as a diagnostic server on the network. Move to the original node, and enter the `net diag` command at the command-line level.

Windows 95 provides fundamental troubleshooting information for wide area networking through its system of Help screens. Just select Help from the Control Panel's toolbar and click on the topics related to the problem. Also use a word processing package to read the Windows 95 SETUPLOG.TXT and BOOTLOG.TXT files. These files record where setup and booting errors occur. Use the F8 function key, during bootup, to examine each driver being loaded.

Novell NetWare and Windows NT

In a client/server system such as a Novell NetWare or Windows NT system, the technician's main responsibility is to get the local station to boot up to the network's logon prompt. At this point, the network administrator or network engineer becomes responsible for directing the troubleshooting process.

In a Novell system, check the root directory of the workstation for the NetBIOS and IPX.COM files. Check the AUTOEXEC.BAT file on the local drive for command lines to run the NetBIOS, load the IPX file, and load the ODI (or NETx) files.

The NetBIOS file is an emulation of IBM's Network Basic Input/Output System, and represents the basic interface between the operating system and the LAN. This function is implemented through ROM BIOS ICs, located on the network card. The Internetworking Packet Exchange (IPX) file passes commands across the network to the file server. The Open Datalink Interface (ODI) file is the network shell that communicates between the adapter and the system's applications. Older versions of NetWare used a shell program called NETx. These files should be referenced in the AUTOEXEC.BAT or NET.BAT files.

In a Windows NT workstation, there are only a few options to get the unit to the logon prompt. If the original installer set the Windows NT client up in FAT mode, you can boot the system up to a DOS prompt. Use a DOS 5.0 or higher boot disk to boot the system. Check for the following files in the root directory:

- NT LOADER (LDR)
- MSDOS.SYS
- COMMAND.COM
- IO.SYS

If the installer used NTFS format to set up the client site, there is no option to boot the system to a DOS prompt. If an Emergency

Repair Disk was created during setup, it can be used to gain access to the system. If no such repair disk is available, however, the Windows NT Client software will need to be reinstalled.

When a Windows NT workstation boots up, the NT LOADER program checks the installed hardware and examines the Windows NT Kernel program. If both checks are correct, a blue screen comes up on the monitor. At this point, you are in the Windows NT operating system. As Windows NT takes over, it loads the interface that includes the Registry and the SERVICES.LPD file. Next, Windows NT brings in the GUI and the logon prompt. If the indicated files are not present, it may be necessary to reinstall Windows NT.

LAN Hardware Checks

Some LAN adapters come with software-diagnostic programs that can be used to isolate problems with the adapter. If this type of software is included, use it to test the card. The diagnostic software can also be used to change the adapter's configuration if necessary.

If the card fails any of the diagnostic tests, check it by exchanging it with a known good one of the same type. Set the replacement card's station address so that it is unique (usually the same as the card being removed). Depending on the type of system being tested, the file server may need to be cycled off and then back on to detect the presence of the new LAN card.

Check the activity light on the back plate of the LAN card (if available) to see whether it is being recognized by the network. If the lights are active, the connection is alive. If not, check the adapter in another node. Check the cabling to make sure that it is the correct type and that the connector is properly attached. A LAN cable tester is an excellent device to have in this situation.

If the operation of the local computer unit appears normal, it will be necessary to troubleshoot the network from the node out. As mentioned earlier, always consult the network administrator before performing any work on a network, beyond a standalone unit. In a network, no node is an island, and every unit has an impact on the operation of the network when it is online. Changes made in one part of a network can cause problems, and data loss, in other parts of the network. You should be aware that changing hardware and software configuration settings for the adapter can have adverse effects, when the system is returned to the network. In addition, changing hard drives in a network node can have a negative impact on the network, when the unit is brought back online.

Check the system for concentrators, routers, and bridges that may not be functioning properly. Check the frame settings being used to make sure that they are compatible from device to device, or that they are represented on the file server. The operation of these devices will have to be verified as separate units.

Network Printing Problems

The reason that network printing problems are located in the LAN section of the “Data Communications” chapter is that the technician must understand the problems that networking brings to the operation before troubleshooting can be effective. As described in Chapter 10, “Printers,” transferring data from the system to the printer over a parallel port and cable is largely a matter of connecting the cable and installing the proper printer driver for the selected printer. The protocol for sending data consists largely of a simple hardware handshake routine. Even in a serial printer, the protocol is only slightly more complex. When a network is involved, however, the complexity becomes that much greater again because of the addition of the network drivers.

The first step in troubleshooting network printer problems is to verify that the local unit and the remote printer are set up for remote printing. In Windows for Workgroups and Windows 95, this involves sharing the printer with the network users. The station that the printer is connected to should appear in the Windows 95 Network Neighborhood window. If the remote computer cannot see files and printers at the print server station, file and print sharing may not be enabled there.

In Windows 95, file and printer sharing can be accomplished at the print server in a number of different ways. In the first method, double-click the printer’s icon in the My Computer or Windows Explorer. Follow this by selecting Printer/Properties/Sharing and then choosing the configuration. The second method calls for an *alternate click* (right-click for right handers) on the printer icon, followed by selecting Share in the Context Menu and choosing the configuration. The final method is similar except that the path is: alternate click/Properties/Sharing/choose the configuration.

The next step is to verify the operation of the printer. As described in the Troubleshooting section of the Printer chapter, run a self-test on the printer to make certain that it is working correctly. Turn the printer off and allow a few seconds for its buffer memory to clear. Try to run a test page to verify the operation of the printer's hardware.

If the test page does not print, there is obviously a problem with the printer. Troubleshoot the printer using the information from Chapter 10 until the self-test works correctly. With the printer working, attempt to print across the network again.

The second step is to determine whether the print server (computer actually connected to the network printer) can print to the printer. Try to open a document on the print server and print it out. If the local printing operation is unsuccessful, move to the MS-DOS prompt, create a small batch file, and copy it to the local LPT port (as described in Chapter 10).

If the file prints, there are a few possible causes of printing problems. The first possibility is that a problem exists with the printer configuration at the print server. Check the print server's drivers.

Another common problem is that there may not be enough memory or hard drive space available in the print server. In Windows 3.x, check the available space on the hard drive and clear the contents of the Temp directory. In Windows 95, check the Spool Settings under the Details entry of the Control Panel/Printers/Properties path. If the spooler is set to EMF, set it to RAW spooling. If the print spool is set to RAW, turn the spool off and click the Print Directly to Printer button. If the unit prints the test page, use the ScanDisK utility to check the disk space. Clear out the Temp directory.

If the file does not print, there is a hardware problem in the local hardware. Refer to the "Hardware Checks" section of the troubleshooting procedure in Chapter 10.

The third step is to verify the operation of the network. This can be accomplished by trying other network functions such as transferring a file from the remote unit to the print sever. If the other network functions work, examine the printer driver configuration of the remote unit. In Windows 3.x, check under Connect/Network in the Control Panel. In Windows 95, open the Control Panel's Printer folder and choose the Properties entry in the drop-down File menu. Check the information under the Details and Sharing tabs.

If the print drivers appear to be correct, install a generic or text-only printer driver and try to print to the print server. Also, move to the DOS prompt in the remote unit and create a batch text file. Attempt to copy this file to the network printer. If the generic driver, or the DOS file works, reinstall the printer driver, or install new drivers for the designated printer.

In the event that other network functions are operational, the final step is to verify the printer operation of the local computer. If possible, connect a printer directly to the local unit and set its print driver up to print to the local printer port. If the file prints to the local printer, a network/printer driver problem still exists. Reload the printer driver and check the network print path. The correct format for the network path is \\COMPUTER_name\SHARED DEVICE.name.Check the network cabling for good connections.

In Windows 95, the Help function will perform many of the diagnostic steps outlined here in a guided format.

If the printer operation stalls or crashes during the printing process, a different type of problem is indicated. In this case, the remote printer was functioning, the print server was operational, and the network was transferring data. Some critical condition must have been reached to stop the printing process. Check the print spooler (or Print Manager) in the print server to see whether an error has occurred. Also, check the hard-disk space and memory usage in the print server.

Wide Area Networks

Objective

The fastest-growing segment of the personal computer world is in the area of `wide area networks` (WANs). A wide area network is very similar in concept to a widely distributed client/server LAN. In a wide area network, computers are typically separated by distances that must be serviced via modems rather than network cards. A typical wide area network would be a local city, or countywide network. This network links network members together through a `bulletin board service` (BBS). Users can access the bulletin board's server with just a telephone call.

Wide area networks are connected by several different types of communication systems. These communication paths are referred to as `links`. Most users connect to the network via standard telephone lines, using dial-up modems like those described earlier in this chapter. Dial-up connections are generally the slowest method of connecting to a network, but they are inexpensive to establish and use.

Other users, who require quicker data transfers, contract with the telephone company to use special, high-speed `Integrated Service Digital Network` (ISDN) lines. These types of links require a digital modem to conduct data transfers. Because the modem is digital, no analog conversion is required.

Users who require very high volumes will lease dedicated `T1 and T3 lines` from the telephone company. These applications generally serve businesses who put several of their computers or networks online. After the information is transmitted, it may be carried over many types of communications links on its way to its destination. These interconnecting links can include fiber-optic cables, satellite up and down links, UHF, and microwave transmission systems.

In some areas, high-speed intermediate-sized networks, referred to as Metropolitan Area Networks (MANs), are popping up. These networks typically cover areas up to 50 kilometers in diameter and are operated to provide access to regional resources. They are like LANs in speed and operation but use special high-speed connections and protocols to increase the geographical span of the network like a WAN.

The Internet

Objective

The most famous wide area network is the `Internet`. The Internet is actually a network of networks, working together. The main communication path for the Internet is a series of networks, established by the U.S. government, to link super computers together at key research sites. This pathway is referred to as the `backbone`, and is affiliated with the National Science Foundation (NSF). Since the original backbone was established, the Internet has expanded around the world, and offers access to computer users in every part of the globe.

TCP/IP

The language of the Internet is Transport Control Protocol/ Internet Protocol, or TCP/IP for short. No matter what type of computer platform or software is being used, the information must move across the Internet in this format. This protocol calls for data to be grouped together in bundles called `network packets`.

The TCP/IP packet is designed primarily to allow for message fragmentation and reassembly. It exists through two `header fields`: the `IP header`, and the `TCP header`, followed by the data field, as illustrated in Figure 11.34.

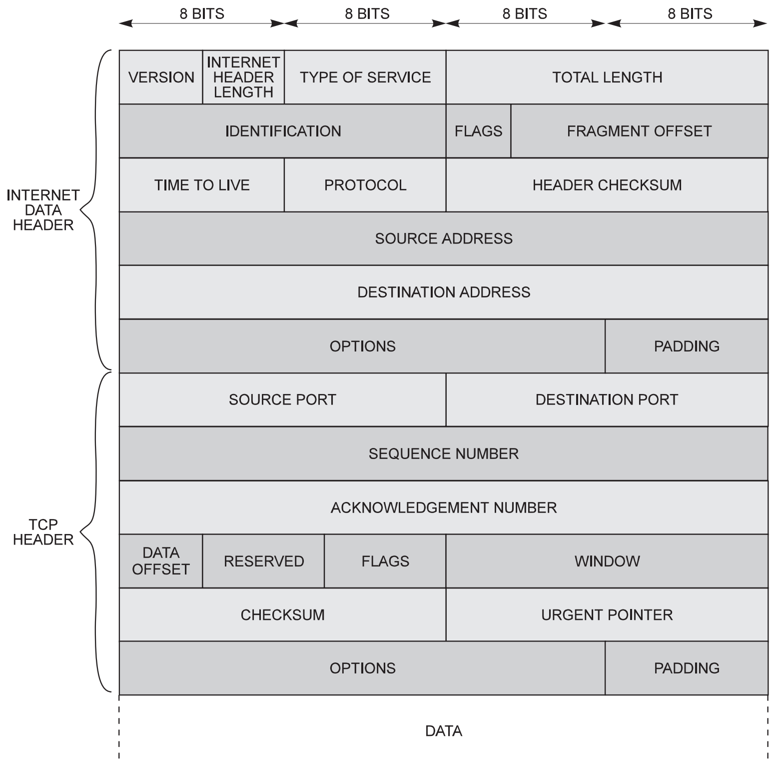


Figure 11.34

TCP/IP packet.

Internet Service Providers (ISPs)

Connecting all the users and individual networks together are `Internet service providers` (ISPs). ISPs are companies that provide the technical gateway to the Internet. These companies own blocks of access addresses that they assign to their customers, to give them an identity on the network. These addresses are called the `Internet protocol addresses`, or `IP addresses`. The IP address makes each site a valid member of the Internet. This is how individual users are identified to receive file transfers, e-mail, and file requests.

IP addresses exist in the numeric format of XXX.YYY.ZZZ.AAA. Each address consists of four 8-bit fields separated by dots. This format of specifying addresses is referred to as `dotted decimal notation`. The decimal numbers are derived from the binary address that the hardware understands. For example, a binary network address of:

10000111.10001011.01001001.00110110 (binary)

corresponds to:

135.139.073.054 (decimal)

Internet Domains

The IP addresses of all the computers attached to the Internet are tracked using a listing system called the `domain name system` (DNS). This system evolved as a method of organizing the members of the Internet into a hierarchical management structure.

The DNS structure consists of various levels of computer groups called `domains`. Each computer on the Internet is assigned a `domain name`, such as `marcraft@oneworld.owt.com`. Each domain name corresponds to an additional domain level.

In the example, the .com notation at the end of the address is a major domain code that identifies the user as a commercial site. The following list identifies the Internet's major domain codes:

* .com = Commercial businesses

* .edu = Educational institutions

* .gov = Government agencies

* .int = International organizations

* .mil = Military establishments

* .net = Networking organizations

* .org = Nonprofit organizations

The .owt identifies the organization that is a domain listed under the major domain heading. Likewise, the .oneworld entry is a subdomain of the .owt domain. It is very likely one of multiple networks supported by .owt. The Marcraft entry is the address location of the end user. If the end user location is an e-mail address, it is usually denoted by an ampersand (@) between its name and the name of its host domain (that is, **mcarft@oneworld.owt.com**).

In each domain level, the members of the domain are responsible for tracking the addresses of the domains on the next lower level. The lower domain is then responsible for tracking the addresses of domains, or end users, on the next level below them.

In addition to its domain name tracking function, the DNS system links the individual domain names to their current IP address listings.

The Internet software communicates with the service provider by embedding the TCP/IP information in a `Point-to-Point Protocol` (PPP) shell for transmission through the modem in analog format. The communications equipment, at the service provider's site, converts the signal back to the digital TCP/IP format. Older units running UNIX used a connection protocol called `Serial Line Internet Protocol` (SLIP) for dial-up services.

Some service providers, such as America Online (AOL) and CompuServe, have become very well known. However, there are thousands of lesser-known, dedicated Internet access provider companies offering services around the world. Figure 11.35 illustrates the service provider's position in the Internet scheme, and shows the various connection methods used to access the Internet.

When you connect to a service provider, you are connecting to their computer system, which in turn is connected to the Internet through devices called routers. A router is a device that intercepts network transmissions and determines which part of the Internet they are intended for. It then determines what the best routing scheme is for delivering the message to its intended address. The routing schedule is devised on the known, available links through the Internet and the amount of traffic detected on various segments. The router then transfers the message to a network access point (NAP).

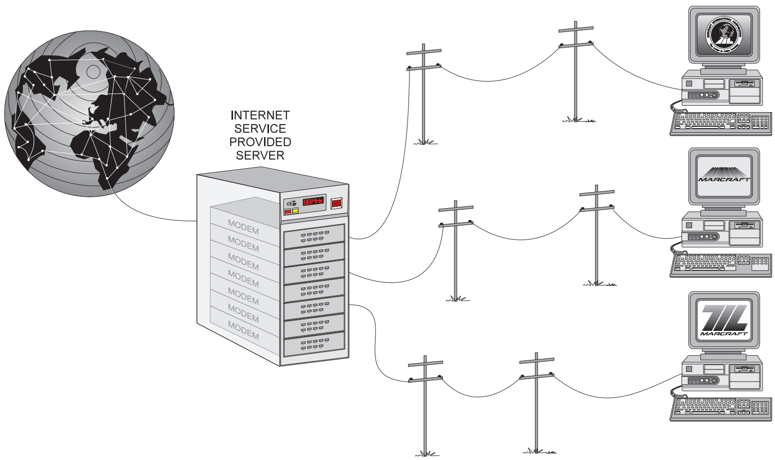


Figure 11.35

Service provider's position.

Internet Transmissions

The TCP/IP protocol divides the transmission into packets of information suitable for retransmission across the Internet. Along the way, the information passes through different networks that are organized at different levels. Depending on the routing scheme, the packets may move through the Internet using different routes to get to the intended address. At the destination, however, the packets are reassembled into the original transmission. This concept is illustrated in Figure 11.36.

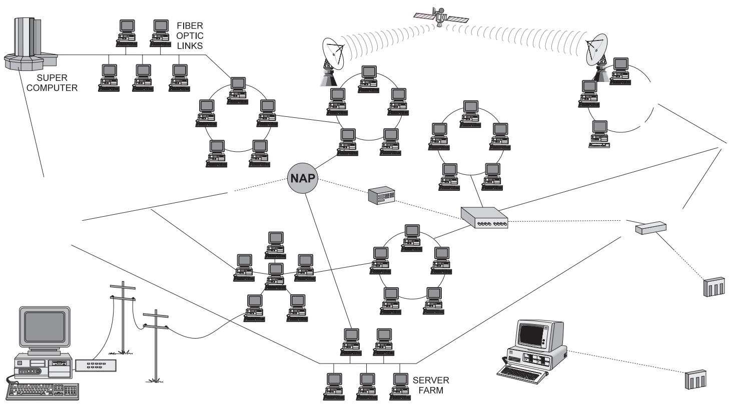


Figure 11.36

Packets moving through the Internet.

As the message moves from the originating address to its destination, it may pass through local area networks, mid-level networks, routers, repeaters, hubs, bridges and gateways. A mid-level network is just another network that does not require an Internet connection to carry out communications. A router receives messages, amplifies them, and retransmits them to keep the messages from deteriorating as they travel. Hubs are used to link networks together so that nodes within them can communicate with one another. Bridges connect networks together so that data can pass through them as it moves from one network to the next. A special type of bridge, called a gateway, translates the message as it passes through so that it can be used by different types of networks (Apple networks and PC networks).

ISDN

Objective

As discussed earlier in the chapter, ISDN service offers high-speed access to the public telephone system. However, ISDN service requires digital modems (also referred to as `terminal adapters` (TAs). Not only does the end user require a digital modem, the telephone company's switch gear equipment must be updated to handle digital switching. This fact has slowed implementation of ISDN services until recently.

Three levels of ISDN service are actually available, Basic Rate Interface (BRI) service, Primary Rate Interface (PRI) services, and Broadband ISDN (BISDN) services.

BRI services are designed to provide residential users with basic digital service through the existing telephone system. The cost of this service is relatively low, although it is more expensive than regular analog service. BRI service is not available in all areas of the country, but it is expanding rapidly.

Typical residential telephone wiring consists of a four-wire cable. Up to seven devices can be connected to these wires. Under the BRI specification, the telephone company delivers three information channels to the residence over a two-wire cable. The two-wire system is expanded into the four-wire system at the residence through a `network terminator`. The ISDN organization structure is depicted in Figure 11.37.

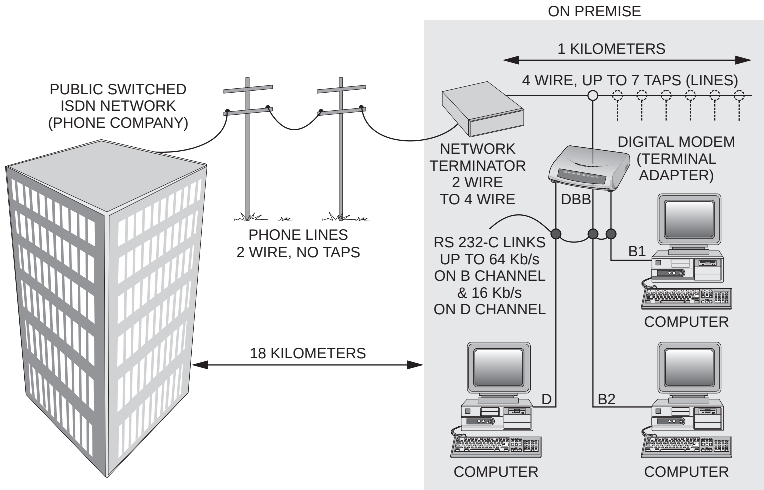


Figure 11.37

ISDN organizational structure.

The BRI information channels exist as a pair of 64Kbps channels and a 16Kbps control channel. The two 64Kbps channels, called `bearer` or `B channels`, can be used to transmit and receive voice and data information. The 16Kbps `D channel` is used to implement advanced control features such as call waiting, call forwarding, Caller ID, and others. The D channel can also be used to conduct packet transfer operations.

PRI services are more elaborate ISDN services that support very high data rates needed for live video transmissions. This is accomplished using the telephone company's existing wiring and advanced ISDN devices. The operating cost of PRI service is considerably more expensive than BRI services. The higher costs of PRI tend to limit its usage to larger businesses.

The fastest, most-expensive ISDN service is Broadband ISDN. This level of service provides extremely high (up to 622Mbps) over coaxial or fiber-optic cabling. Advanced transmission protocols are also used to implement BISDN.

Digital modems are available in both internal and external formats. In the case of external devices, the analog link between the computer and the modem requires a D-to-A and A-to-D conversion processes at the computer's serial port and then again at the modem. Of course, with an internal digital modem these conversion processes are not required.

File Transfer Protocol

Objective

A special application, called the `File Transfer Protocol` (FTP), is used to upload and download information to and from the Internet. FTP is a client/server type of software application. The server version runs on the host computer; the client version runs on the user's station. To access an FTP site, the user must move into an FTP application and enter the address of the site to be accessed. After the physical connection has been made, the user must log on to the FTP site by supplying an account number and password. When the host receives a valid password, a communication path opens between the host and the user site, and an `FTP session` begins.

Around the world, thousands of FTP host sites contain millions of pages of information that can be downloaded free of charge. Special servers, called `Archie servers` (archival servers), contain listings to assist users in locating specific topics stored at FTP sites around the world. Another network information gathering/file management utility is `Gopher`. Gopher is an Internet tool developed at the University of Minnesota that enables users to `search` (find computers that have informational topics of interest) and `browse` (look through information at various sites).

In the Windows 3.x environment, a driver that follows the `Windows Sockets 1.1` specification is used to interface the Windows operating environment to the TCP/IP protocol. Likewise, a Windows gopher, such as `Hgopher`, can be loaded so that Windows applications can communicate directly with the Internet protocol.

E-Mail

Objective

One of the most widely used functions of wide area networks is the `electronic-mail` (e-mail) feature. This feature enables Internet users to send and receive electronic messages to one another over the Internet. As with the regular postal service, e-mail is sent to an address from an address. With e-mail, however, you can send the same message to several addresses at the same time, using a mailing list.

Several e-mail programs are available to provide this function. E-mail is normally written as ASCII text files. These files can be created using an ordinary word processing package. An e-mail `mailer` program is then used to drop the text file into an electronic mailbox. E-mail can also have graphics, audio, and files from other applications `attached` to them. To run them, however, the intended user will need to have the same application packages that were originally used to created the files.

On the Internet, the message is distributed into packets, as with any other TCP/IP files. At the receiving end, the e-mail message is reassembled and stored in the recipient's mail box. When the designated user boots up on the system, the e-mail program delivers the message and notifies the user that it has arrived. The user can activate the e-mail `reader` portion of the program to view the information.

The World Wide Web

Objective

The `World Wide Web` (WWW) is a menu system that ties together Internet resources from around the world. These resources are scattered across computer systems everywhere. `Web servers` inventory the Web's resources and store address pointers (links) to them. These links are used to create `hypermedia` documents that can contain information from computer sites around the world. Inside a hypermedia document, the links enable the user to move around the document in a nonlinear manner. In an online encyclopedia, for example, the user can move around the encyclopedia to review all the entries concerning a single topic, without reading through every entry looking for them. The contents of the document can be mixed as well. A hypermedia document may contain text, graphics, and animation, as well as audio and video sequences.

Each Web site has a unique address called its `Uniform Resource Locator` (URL). URLs have a format similar to a DOS command line. To access a Web site, the user must place the desired URL on the network. Each URL begins with the characters `http://`. These characters stand for Hypertext Transfer Protocol, and identify the address as a Web site. The rest of the address is the name of the site being accessed (for example, **http://www.mic-inc.com** is the homepage of Marcraft, located on a server at One World Telecommunications). Each Web site begins with a `home page`. The home page is the menu to the available contents of the site.

Web Browsers

Objective

As the Internet network has grown, service providers have continued to provide more user-friendly software for exploring the World Wide Web. These software packages are called browsers, and are based on `hypertext links`. Browsers use hypertext links to interconnect the various computing sites in a way that resembles a spider's web—hence the name Web.

Browsers are to the Internet what Windows is to operating environments. Graphical browsers such as Mosaic, Netscape Navigator, and Microsoft Internet Explorer, enable users to move around the Internet and make selections from graphically designed pages and menus instead of operating from a command line. The original Internet operating environment was a command-line program called UNIX. Fortunately, the UNIX structure and many of its commands were the basis used to create MS-DOS. Therefore, users that are DOS-literate do not require extensive training to begin using UNIX. However, with the advent of a variety of browsers, it is unlikely that most users will become involved with UNIX.

The National Center for Supercomputing Applications introduced the first graphical browser in 1993. This program was known as Mosaic. As its name implies, Mosaic allowed graphical pages to be created using a mixture of text, graphics, audio, and video files. It translated the `Hypertext Markup Language` (HTML) files that were used to create the Web and that ultimately link the various types of files together.

Mosaic was soon followed by Netscape Navigator and the Microsoft Internet Explorer. Figure 11.38 depicts the home page (presentation screen) for the Netscape Navigator from Netscape Communications Corporation.

Figure 11.38

Netscape Navigator home page.

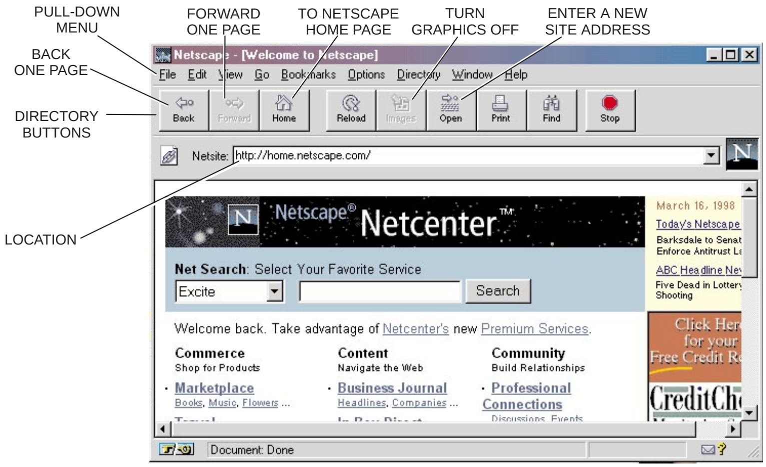


Figure 11.39 illustrates the Microsoft Internet Explorer. Its features are similar to those of the Netscape Navigator. Both provide a graphical interface for viewing Web pages. Links to search engines are useful for finding information on the Internet. Both have links to built-in, e-mail facilities and to their respective creators' home pages. In Netscape Navigator, searches look at Netscape-recommended sites; the Explorer, on the other hand, first checks out Microsoft sites. Operating either browser in Windows versions, before Windows 95, requires an external windows socket program to be loaded before running the browser. With Windows 95, the socket was designed into the operating environment.

Figure 11.39

Internet Explorer home page.

Several software packages enable users to generate their own Web pages. Programs, such as word processors and desktop publishers, have included provisions for creating and saving HTML files, called `applets`, that can be used as home pages. Internet browsers, such as Netscape and Internet Explorer, include facilities for generating home page documents. `Scripting languages`, such as Java, are also used to create HTML applets.

Troubleshooting WANs

Objective

Unless you work for an Internet service provider, most of the work at an Internet site involves the components and software of the local computer. Most of the troubleshooting from the local computer level is identical to the steps given earlier for the modem. Run the loopback tests and try to get online with a local dial-up service by using a simple communications software package (such as Terminal). Check the spelling of IP addresses to make sure they are spelled exactly as they should be. If the spelling is wrong, no communications will take place. The major difference in checking wide area network problems occurs in checking the Internet-specific software, such as the browser.

Checking the modem or network card is the major hardware-related activity normally involved with Internet sites. However, you may be called on to work with the customer's local Internet service provider to solve some problems. Each user should have received a packet of information from his service provider when the service was purchased. These documents normally contain all the ISP-specific configuration information needed to set up the user's site. This information should be consulted when installing and configuring any Internet-related software.

Windows 95 provides fundamental troubleshooting information for wide area networking through its system of Help screens. Just select Help from the Control Panel's toolbar, and then click on the topic that you are troubleshooting.

Summary

This chapter has investigated the two major areas of data communications associated with personal computer systems. These areas include local area networks (LANs) and wide area networks (WANs).

The opening sections of the chapter looked at modems. This material is a natural addition to the I/O port information delivered in Chapter 7, “Input/Output.” As a matter of fact, both sections should be covered when troubleshooting serial communications problems. Modems are the most widely used data communication devices in use today. The operation and troubleshooting of modems were presented in this section of the chapter.

In the LANs portion of the chapter, the major local area network topologies were described. These included Ethernet, ARCnet, and Token Ring LANs. Basic LAN operation and troubleshooting procedures were covered in depth.

The WAN section of the chapter moved quickly into subjects surrounding the Internet. These subjects included File Transfer Protocol (FTP), e-mail, the World Wide Web, and browsers. As with all the other major section of this text, a troubleshooting section occurred at the end of the section to cap it off.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each point and that you can perform each task listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercises

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 31 - Windows 3.x Terminal, 32 - Modem Installation and Setup, 33 - Network Installation and Setup, and 34 - Windows Internet Explorer.

Review Questions

1. What is the minimum number of PCs that need to be connected together before a true network is formed?
2. What type of topology is Ethernet?
3. To which topology does ARCnet belong?
4. Describe the function of the World Wide Web.
5. Describe synchronous transmission versus asynchronous transmission.
6. What is the primary difference between a client/server type of network and a peer-to-peer network?
7. Describe the three important configuration settings associated with a network adapter card.
8. What element determines the active unit in a Token-Ring network?
9. State the maximum segment length of a 10base2 Ethernet network.
10. List three possible places the UART circuitry can be located in a PC-compatible system.
11. What is the purpose of a router in a network?
12. Describe the type of conductor used in a 1base5 network.
13. What is TCP/IP, and what does it mean?

14. What function does a browser perform?
15. How does a Token-Ring network keep a single unit from dominating the network after it receives the token? What is this called?
16. What is the first step in checking a networked computer?
17. Define the word `protocol`.
18. Describe the tests that a typical modem can perform on itself.
19. What is the bandwidth of the telephone system?
20. How are baud rate and bit rate different?
21. To which type of communications products do Hayes-compatible commands pertain?
22. Under what conditions is a modem required to transmit data?
23. Define the acronym URL, and describe what it is used for.
24. What language is used to create documents for the World Wide Web?
25. Describe the function of an Internet service provider.

Review Answers

1. Three. With only two computers, a point-to-point communication link is established, not a network. In a network, an additional level of protocol is required to differentiate among possible receivers on the network. For more information, see the section titled “Local Area Networks.”
2. A bus topology. For more information, see the section titled “Ethernet Specifications.”
3. A ring topology. For more information, see the section titled “ARCnet Specifications.”

4. A menu system that ties together the resources of the Internet. For more information, see the section titled "The World Wide Web."

5. Synchronous transmissions send data at regular intervals with timing information included in the transmission. Asynchronous transmission techniques send characters at irregular intervals marked by start and stop bits. For more information, see the section titled "Transmission Synchronization."

6. Client/server systems use centrally located computers to handle programs and data for the entire system. Peer-to-peer networks connect otherwise autonomous stations together so that they can share information and resources. For more information, see the section titled "LAN Topologies."

7. IRQ channel, base I/O port address, and base memory address settings. For more information, see the section titled "Installing LANs."

8. Possession of the network's token. For more information, see the section titled "ARCnet."

9. 185 meters. For more information, see the section titled "Ethernet Specifications."

10. The UART can be located in the modem card of an internal modem, on the MI/O card, or in an ASIC on the system board. For more information, see the section titled "The Serial Interface."

11. Receives network traffic, amplifies it, and then resends it so that it does not deteriorate during transmission. For more information, see the section titled "Internet Domains."

12. UTP cable. For more information, see the section titled "Ethernet Specifications."

13. Transport Control Protocol/Internet Protocol. This is the language of the Internet. For more information, see the section titled "LAN Adapter Cards."

14. Browsers enable users to move around the Internet and make selections from graphically designed pages and menus instead of operating from a command line. For more information, see the section titled “Web Browsers.”

15. The protocol only allows the node to hold the packet for a predetermined amount of time. For more information, see the section titled “ARCnet.”

16. Determine what has changed since the system was run last. If the unit is a new installation, treat it as a setup problem. For more information, see the section titled “Troubleshooting LANs.”

17. An agreed-upon method of doing things. In a data transmission system, it is a set of rules for transferring information. For more information, see the section titled “Protocols.”

18. The local digital loopback test checks the serial port circuitry; the local analog loopback test checks the local serial port and the local modem; the remote digital loopback test checks the local port, the local modem, the transmission line, and the remote modem. For more information, see the section titled “Modem Hardware Checks.”

19. 3,000 Hz. For more information, see the section titled “Understanding How Modems Work.”

20. Baud rate specifies the number of times the electrical signal on the conductor changes while bit rate indicates the number of bits of information that those changes represent. For more information, see the section titled “Understanding How Modems Work.”

21. Modems. For more information, see the section titled “Communication Software.”

22. For serial transmissions greater than 100 feet. For more information, see the section titled “Modems.”

23. Uniform Resource Locator, the address assigned to a Web site on the Internet. For more information, see the section titled "The World Wide Web."

24. Hypertext Markup Language (HTML). For more information, see the section titled "Web Browsers."

25. Organizations that provide the technical gateway to the Internet. For more information, see the section titled "Internet Service Providers (ISPs)."

Chapter 12

Multimedia

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- Define the term `multimedia`.
- Describe how audio signals are digitized.
- Describe how video signals are digitized.
- List file extensions used with different types of multimedia files.
- Describe the function of multimedia authoring programs.
- List hardware components associated with various multimedia configurations.
- Describe the operation of a writable CD drive.
- Describe steps to troubleshoot CD-ROM problems.
- Describe steps to troubleshoot sound card problems.
- Describe steps to troubleshoot video capture card problems.
- State the characteristics of MPEG and JPEG compression formats.
- Describe steps to troubleshoot VGA-TV card problems.
- List specifications for different levels of multimedia systems.

continues

- Differentiate between different types of CDs.
- Describe the MIDI standard and discuss where it is used.
- Install and configure a sound card for operation.
- Install and configure a video capture card for operation.
- Install and configure a CD-ROM drive for operation.
- Install and configure a VGA-TV adapter for operation.

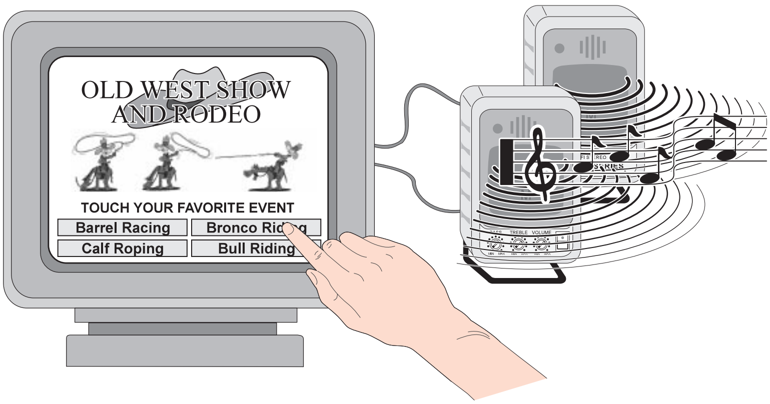
Introduction

The fastest-growing application of microcomputers is in the area of multimedia creation and presentation. Multimedia is a term applied to a range of applications that bring together text, graphics, video, audio, and animation to provide interactivity between the computer and its human operator. This concept is illustrated in Figure 12.1. Typical applications for multimedia include interactive presentations, computer-based and computer-assisted instruction materials, and interactive customer-service centers (kiosks).

Computer-based instruction (CBI) applies to instructional materials where the computer is the main source of instructional content. Computer-assisted instruction (CAI) is a term used when the computer is employed as a presentation assistant and class-management tool. A kiosk is a computer system set up to advise users about some topic. Large grocery and hardware stores use kiosks to direct customers to the locations of various types of merchandise available in the store.

Figure 12.1

Multimedia interactivity.



Multimedia Applications

Objective

Several different types of software packages are involved in creating all the pieces that go together to make a multimedia title. Graphic-design programs are generally used to create the artwork and other graphic elements to be included in the title. Types of artwork employed in a multimedia project run from designing backdrops and scenery to creating individual picture elements. Special-effects graphics packages or graphics-design packages with special-effect features are used to produce special effects such as animation and morphing.

Video-capture software is used to capture frames of television video and convert them into digital formats that can be processed by the system. Graphics packages may be used to manipulate the contents of the video after it has been converted into digital formats that the computer can handle. One of the most popular file formats for video is the Microsoft Audio Visual Interface (AVI) format.

Audio digitizing software converts music, voice, and sound effects into formats that can be included in the presentation. These packages typically include utilities that enable the producer to edit and modify the sound files created. The most popular sound formats in use are the Musical Instrument Digital Interface (MIDI) and the Wave (WAV) audio file formats. The MIDI format was developed to allow electronic musical instruments and controllers to communicate with each other.

A word processor program may be used to prepare text materials for the screens in the production. The text used in multimedia production is normally created as ASCII files. These files can be used just as text files or converted into a graphic format. Typical graphics formats used in multimedia systems include bitmapped graphics, as well as two- and three-dimensional renderings.

The elements of the multimedia title are brought together, organized, and finished by a type of software package known as an `authoring system`. In addition to combining the elements into a cohesive presentation, the authoring system provides the tools with which to create `interactivity`. Interactivity refers to the ability of the user to participate with the program while it is running, and alter its action. In many cases, the programmer creates scenarios that enable the intended user of the program to select pathways through the software or sections of the software to be used, or to answer questions included in the software.

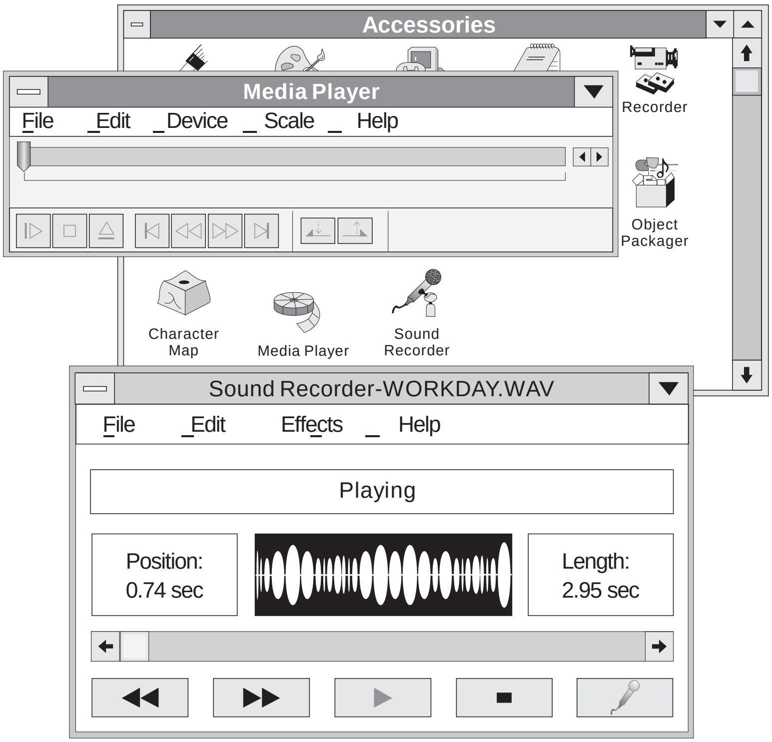
Windows Multimedia Support

Objective

Microsoft Windows provides a fundamental set of multimedia tools for the personal computer. These tools are reached using the Sound Recorder and Media Player icons in the Accessories window. These utilities, depicted in Figure 12.2, operate in conjunction with the settings in the Driver section of the Windows Control Panel. In addition to the tools mentioned, Windows adds a number of file extension specifications to deal with multimedia applications. The primary multimedia extensions are the RIFF and MIDI file formats. The RIFF (Resource Interchange File Format) format supplies a standard file format for graphics, animation, and audio. The MIDI format is the standard for recording and playback of digital musical instruments.

Figure 12.2

Windows multimedia utilities.

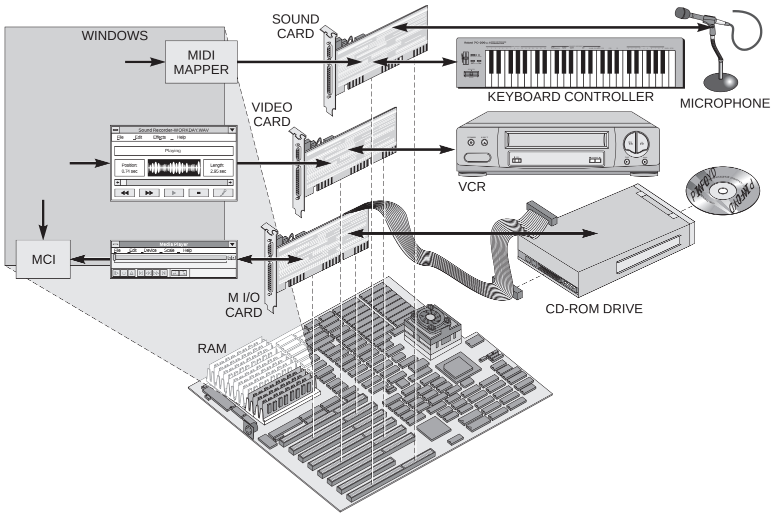


Media Control Interface (MCI)

Windows also supports a software interface specification that allows a Windows-based computer to control external multimedia devices such as VCRs and laser-disc systems. This interface is referred to as the Media Control Interface (MCI). In addition to controlling external devices, the MCI provides a method for Windows-based machines to save or play audio files with its Sound Recorder; play audio, video, or MIDI files from its Media Player; and select various musical instrument sounds using the MIDI Mapper. The functional positions of these utilities are illustrated in Figure 12.3.

Figure 12.3

Windows multimedia utilities in operation.



Audio Tools

The Sound Recorder utility allows Windows to communicate with sound adapter cards to record and play audio in the form of WAV files. The Sound Recorder interface uses start, stop, reverse, fast-forward, fast-reverse, and play icons like those found on a commercially available audio tape recorder.

The Media Player uses a button scheme that corresponds to the transport control buttons on a VCR machine. It is the main channel to the MCI.

The MIDI interface specification was designed to let electronic music devices communicate with each other and with a computer. The MIDI Mapper directs MIDI files to the MIDI instruments they are associated with. A MIDI system can contain several electronic instruments that play in conjunction with the computer system.

Video Tools

Windows does not provide any direct support for handling or playing digitized video files. There are add-on programs for Windows, such as Microsoft's Video for Windows, that can be implemented to play Audio Visual Interface (AVI) files from Windows. This add-on brings several important multimedia-related utilities to the Windows structure. These utilities include the following functions:

- **VidCap.** Works with video and sound capture cards to digitize audio and video clips for Windows. In addition, VidCap allows the video clip to be run in different sizes and speeds. The other utilities are programs that can be used to edit the video and audio clips.
- **VidEdit.** Used to cut and paste video segments together.
- **WavEdit.** Provides the tools to edit the audio portion of the AVI file.
- **PalEdit and BitEdit.** Used to enhance the video by manipulating the color palette and correcting images within the frames of the video clip.

Windows 95 Multimedia Support

In Windows 95, the multimedia tools are located under the Control Panel's Multimedia icon, as illustrated in Figure 12.4. The multimedia tools contained in the icon include Audio, Video, MIDI, CD Music, and Advanced Properties. The Audio tab provides the interface that allows the playback driver and volume value to be set. This tab also handles the audio recording driver setting, internal volume control, and quality settings.

The Video tab allows a user to adjust the window size for playing video files on the screen. Typical settings include original size, fractional portions of the screen, and full screen presentation. The onscreen preview window shows the type of video reproduction that can be expected using the selected window size setting.

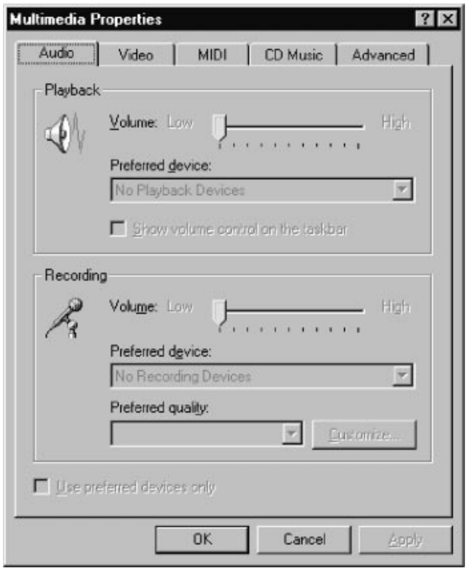


Figure 12.4

Windows 95 multimedia properties.

The MIDI tab allows the system's MIDI drivers to be configured, as well as allowing custom MIDI configurations to be created. Behind the CD Music tab, the CD-ROM drive's settings are established. The relative volume level for the CD-ROM output is defined here.

The different drivers available for the various multimedia tools in the system can be viewed under the Advanced tab. Figure 12.5 illustrates the Advanced Multimedia Properties dialog box for a given system.

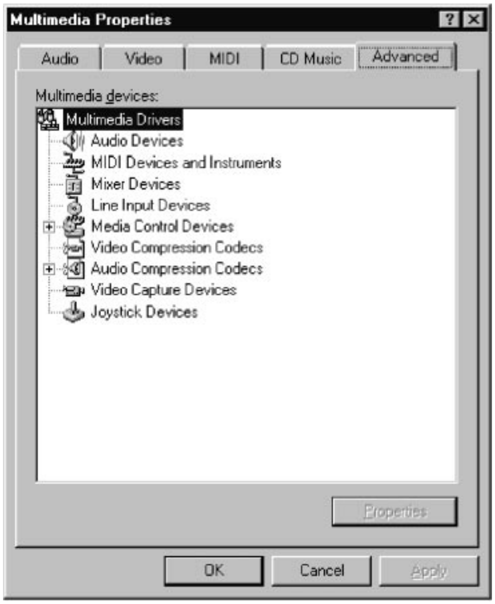


Figure 12.5

Windows 95 multimedia tools.

Authoring Systems

Objective

At the heart of the multimedia movement are the authoring systems that allow various media types to be pulled together into a complete package. Figure 12.6 illustrates the relationship between various media sources and the computer hardware.

Designing Multimedia Titles

Unlike word processing and desktop publishing applications, which are used to produce a static page, multimedia authoring programs are used to prepare time-sequenced and interactive presentations. These programs bring together production techniques associated with radio and video production. To use them to their full potential, the multimedia programmer must think like TV and radio producers. Indeed, multimedia programmers borrow presentation techniques used by people in these fields. It is very common for a multimedia programmer to begin work on a multimedia title by producing a `story board`, as shown in Figure 12.7. A story board is a paper description of the various screens that will be developed, and a narrative of how the events in the screens will play out. In addition, the programmer may develop a play list of audio background or voice clips that will be used in the work.

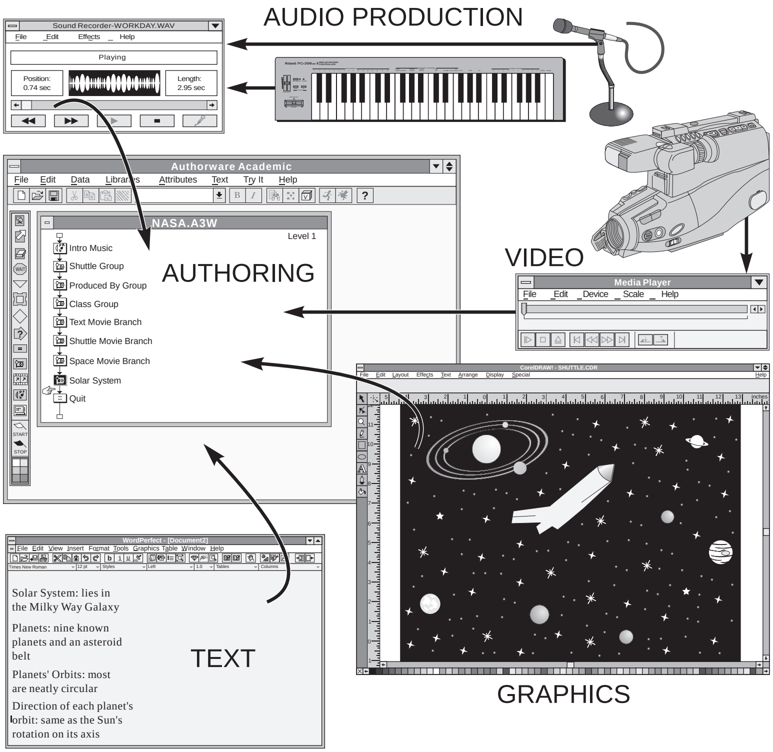


Figure 12.6

Elements of a multimedia title.

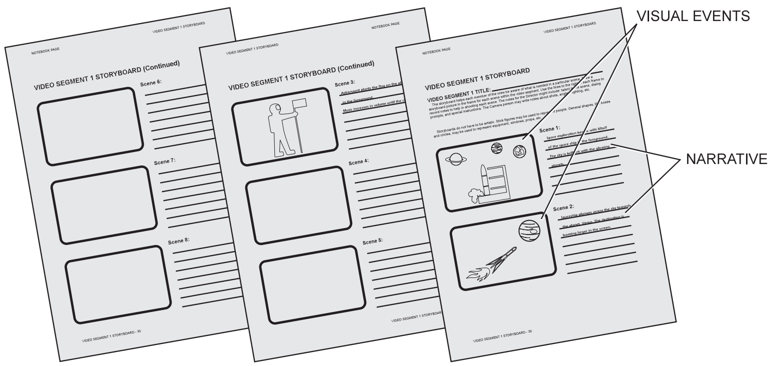


Figure 12.7

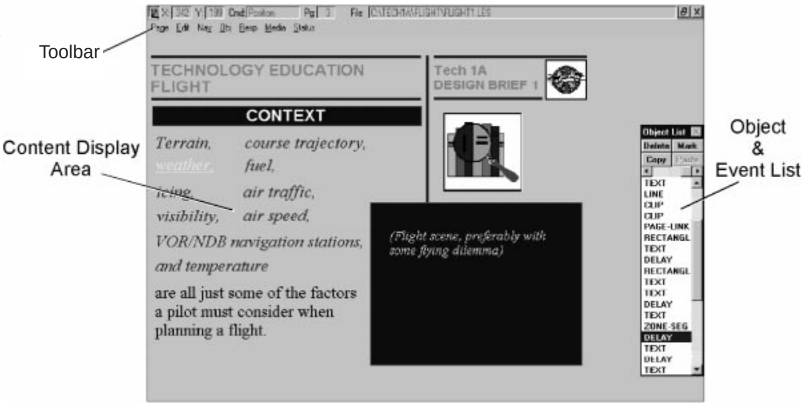
Story board sheets.

Multimedia Authoring Tools

It might be easiest to visualize a multimedia production as a stack of screens that can be flipped through. Each page of the stack uses graphic-design composition techniques to maximize its effectiveness. Figure 12.8 illustrates such an authoring system. The program provides a toolbar containing a set of tools with various definitions. The tools are used one at a time to create objects onscreen, such as geometric shapes, text, and colors. Other tools enable the programmer to create time and event delays between activities. Still other tools enable the programmer to load bitmapped graphics into the page, move different objects around the screen, and cut and paste objects onscreen. Each time a tool is used, an activity and its parameters are logged in the page's object list. When the program is run, the author's Runtime Module just calls up the activities from the object list one at a time, and carries them out as they were defined.

Figure 12.8

Object and event-list multimedia programming.



Other authoring programs are more graphical in nature. The programming interface illustrates a flowchart-based method of arranging objects and events in a title. The programmer plots the logical flow of the presentation by linking multimedia pieces together in a flowchart. In this manner, the programmer can have multiple activities occurring on the screen simultaneously, by placing them in parallel with each other. Objects connected in series with each other happen in sequence, from top to bottom of the flowchart.

After the concept, design, and layout work has been completed, the next step in producing a multimedia title is to collect the different file types together so that they are convenient. This could mean importing all the files into a single drive or having all the files available within a single, local network.

Next, the multimedia programmer uses the authoring program's tools to bring the files together in the correct sequence. The activities within the title are dynamic. The programmer must be aware of how the screen will look at each instant of time. Figure 12.9 demonstrates how the multimedia programmer brings together the parts and pieces of the title onscreen.

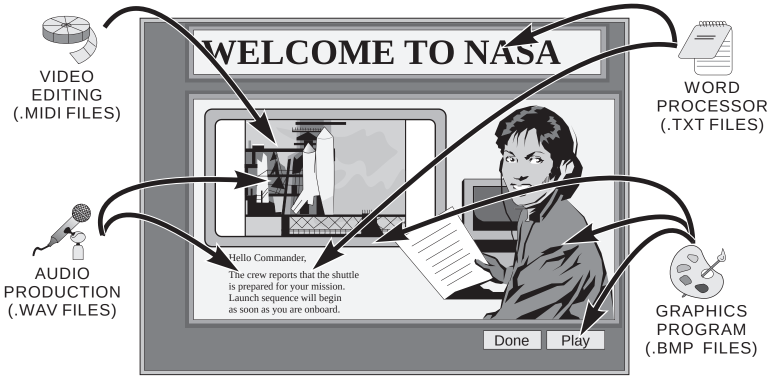


Figure 12.9

Arranging objects and events in a title.

The programmer uses time delays, animated sequences, and video clips to move the user through the presentation. The programmer also injects interactivity by placing menus with buttons on the screen that the user can press to answer questions, get specific information, or alter how the program advances.

The programmer may also include voice clips to narrate the presentation, as well as use music to accompany or highlight the material onscreen. The authoring program is responsible for generating the code necessary to present all the multimedia pieces in the title, at the correct time, during the presentation.

After the title has been completed, it may be placed on a number of different, standard media for distribution. These media include CD-ROM disks, interactive laser discs, video CDs, and networks. A section of the authoring program, called the Runtime Module, is included with the title to make it run on other computers. Most multimedia authoring tools are relatively expensive. To include the entire program, along with a multimedia title, would be costly and wasteful. Therefore, only the Runtime Module required to play back the presentation is normally included with the title.

Multimedia Hardware

Objective

Figure 12.10 depicts a typical multimedia system. Some vendors promote a sound card, a pair of speakers, and a CD-ROM drive as a full multimedia package. This is barely enough capability for a system to be called a multimedia system. PCs intended for multimedia use normally possess very fast microprocessors, large amounts of installed RAM, big, fast hard-disk drives, a fast CD-ROM drive, a sound card, and a high-resolution video card. They may also include items such as still-frame or full-motion video capture cards, VGA-to-TV converter cards, and digital sound cards.

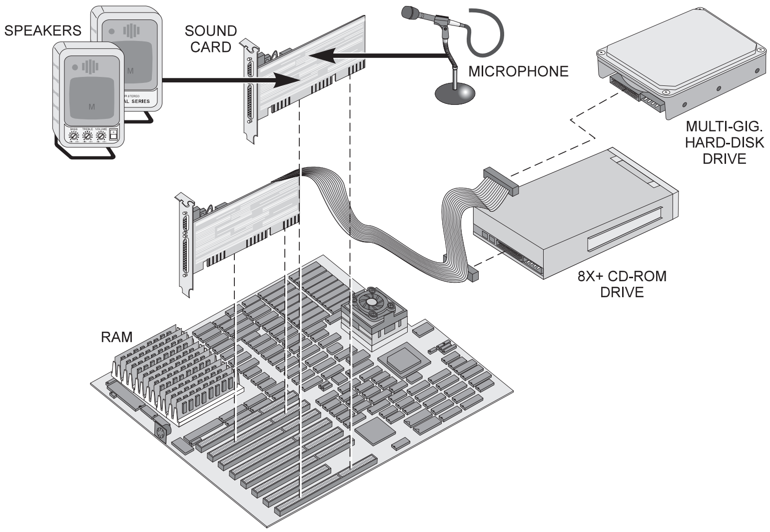


Figure 12.10

A typical multimedia system.

These items are necessary to efficiently bring together text, graphics, sound, animation, and video into a usable package. The large and fast peripheral equipment associated with multimedia stems from the fact that graphic and animation files tend to be large and require a tremendous amount of time to be manipulated. Digitized sound files also tend to be large. Several seconds of simple speech can take up several kilobytes of memory to process and disk space to store. Full-motion video requires an enormous amount of RAM and disk space to manipulate and store. A few seconds of uncompressed, full-motion video can fill a small hard-disk drive.

The high-speed peripheral equipment is required for multimedia applications because of the sheer volume of data processing required to move sound and video files around the system (that is, to and from the hard drive, to the video memory, and on to the screen). In addition to large hard-disk drives, drive arrays are often included in multimedia systems to hold and help process all the data used to create a serious multimedia title.

A standard has been established to define exactly what makes up a minimal-level multimedia computer. This standard is referred to as the Multimedia Personal Computer (MPC) standard. The first MPC level (MPC Level 1) never gained widespread support because of its lack of true multimedia support components. The MPC Level 2 specification calls for a minimum 80486SX-25 or better microprocessor, with at least 4MB of RAM, a 150MB or larger hard drive, a 2X or better CD-ROM drive, a 16-bit sound card, and a 640×480 VGA monitor with 64KB color capabilities.

A high-end MPC system, such as the one in Figure 12.11, is normally built around a Pentium MMX microprocessor, with 64MB or more of RAM. The system board should have a large cache and multiple 32-bit PCI expansion slots. The hard drive should be in the multiple gigabyte range, and an 8X CD-ROM drive should be included. Additional data storage should be available in the form of a tape backup system, a drive array, or one of several removable storage systems.

Audio/video add-ons include a video capture card, a 16-bit sound card, and a 24-bit graphic accelerator video card. A 17-inch SVGA monitor for viewing the work is a desirable addition, and a speaker and microphone round out the audio/video options. A flatbed scanner can be a valuable tool for converting existing pictures and artwork into electronic files that can be used by the MPC system. Finally, a modem enables the author to gather and distribute multimedia material without leaving the digital world. The Internet is certainly a valuable asset to the multimedia programmer.

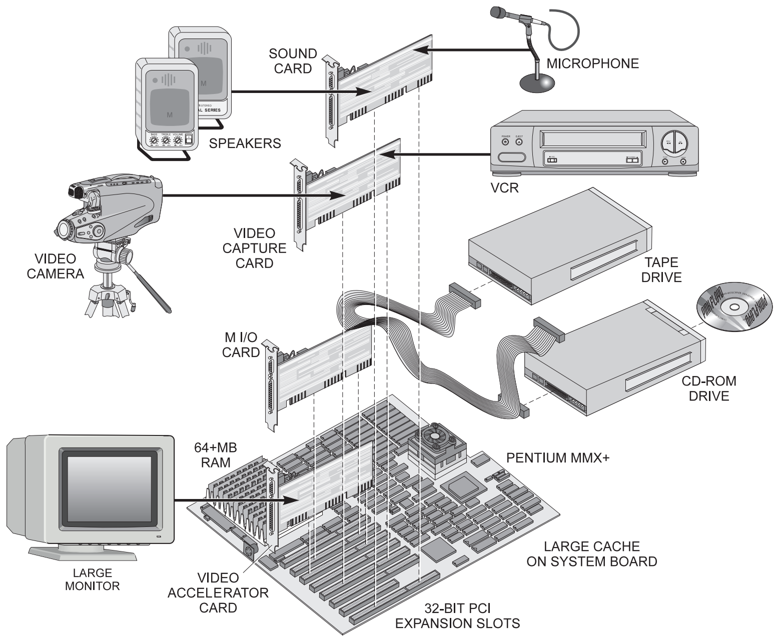


Figure 12.11

A high-end multimedia system.

With all of the hardware options that can be combined into a multimedia unit, it will be necessary to consider the number of system resources that may be required. In particular, the number of IRQ and DMA channels available should be taken into account when considering upgrade options. The number and type of expansions slot available is another point for consideration. For example, it may be necessary to find a scanner that will work off of the existing parallel port, rather than a unit that has a proprietary adapter card. There may not be enough slots to handle the additional peripheral.

CD-ROM

Soon after compact disks (CDs) became popular for storing audio signals on optical material, the benefits of storing computer information in this manner became apparent. With a CD, data is written digitally on a light-sensitive material by a powerful, highly focused laser beam. The writing laser is pulsed by the modulated data to be stored on the disk. When the laser is pulsed, a microscopic blister is burned into the optical material, causing it to reflect light differently from the material around it. The blistered areas are referred to as `pits`; the areas between them are called `lands`.

The recorded data is read from the disk by scanning it with a lower-power, continuous laser beam. The laser diode emits the highly focused, narrow beam that is reflected back from the disk. The reflected beam passes through a prism and is bent 90 degrees, where it is picked up by the diode detector and converted into an electrical signal. Only the light reflected from a land on the disk is picked up by the detector. Light that strikes a pit is scattered and is not detected. The lower power level used for reading the disk ensures that the optical material is not affected during the read operation.

With an audio CD, the digital data retrieved from the disk is passed through a digital-to-analog converter (DAC) to reproduce the audio sound wave. This is `not` required for digital computer systems, however, because the information is already in a form acceptable to the computer. Therefore, CD players designed for use in computer systems are referred to as CD-ROM drives, to differentiate them from audio CD players. Otherwise, the mechanics of operation are very similar between the two devices. The ROM designation refers to the fact that most of the drives available are read-only.

CD-ROM Disks

A typical CD-ROM disk is 4.7 inches in diameter, and consists of three major parts:

- Acrylic substrate
- Aluminized, mirror-finished data surface
- Lacquer coating

The scanning laser beam comes up through the disk, strikes the aluminized data surface, and is reflected back. Because there is no physical contact between the reading mechanism and the disk, the disk never wears out. This is one of the main advantages of the CD system. The blisters on the data surface are typically just under 1 micrometer in length, and the tracks are 1.6 micrometers apart. The data is encoded by the length and spacing of the blisters and the lands between them.

The information on a compact disk is stored in one, continuous spiral track, unlike floppy disks, where the data is stored in multiple, concentric tracks. The compact disk storage format still divides the data into separate sectors. However, the sectors of a CD-ROM disk are physically the same size. The disk spins counterclockwise and slows down as the laser diode emitter/detector unit approaches the outside of the disk. It begins spinning at approximately 500 rpm at the inner edge of the disk, and slows down to about 200 rpm at the outer edge of the disk. The spindle motor controls the speed of the disk so that the track is always passing the laser at between 3.95 and 4.6 feet per second. Therefore, CD-ROM drives must have a variable-speed spindle motor, and cannot just be turned on and off like a floppy drive's spindle motor. The variable speed of the drive allows the disk to contain more sectors, thereby giving it a much larger storage capacity. In fact, the average storage capacity of a CD-ROM disk is about 680MB.

CD-ROM Drives

CD-ROM drives that operate at the speed of a conventional audio CD player are called single-speed drives. Advanced drives that spin twice, and three times, as fast as the typical CD player are referred to as double-speed drives, triple-speed drives, and so on. Single-speed drives transfer data at a rate of 150KB per second. Double-speed drive transfers occur at 300KB per second, and so on.

CD-ROM drives are capable of playing audio CDs. However, a CD player will not be able to produce any output from the CD-ROM disk. CDs are classified by a color-coding system that corresponds to their intended use. CDs that contains digital data intended for use in a computer are referred to as Yellow Book CDs. Red Book

CDs refer to those formatted to contain digital music. Orange Book refers to the standard for CDs used in WORM drives (described later). Green Book CDs are used with interactive CD systems, and Blue Book CDs are those associated with laser-disc systems.

Common applications for CD-ROM drives include reference material such as the following:

- Dictionaries
- Encyclopedias
- Maps
- Graphics for desktop publishing
- Multimedia
- Audio/video clips
- Games for entertainment

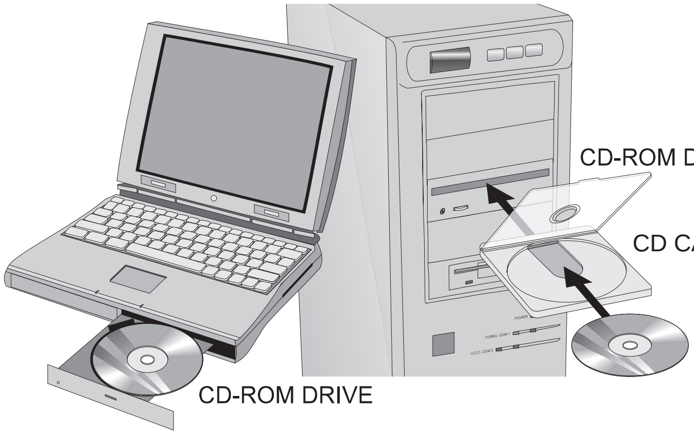
Older hard drives were a bit slow for the demands of multimedia, but as technology improved, performance soared. The original CD-ROM drives had a transfer rate of 150KB/second. Some time after, double-spin (2X) drives began to appear with a 300KB/second transfer rate. Soon after, a few manufacturers began producing a triple-spin (3X) drive; most manufacturers, however, focused on a quad-spin (4X) drive. There are a few 6X drives, but most manufacturers are now focusing on 8X and 12X drives.

There are two common methods for inserting CDs in the drive. The first method involves inserting the CD into a protective caddie, which, in turn, is inserted into the drive. The CD turns inside the caddie, which acts to properly align the disc in the drive.

With the second method, the CD is placed in a shelf that pops out of the drive. Pressing a small button on the front of the drive makes the shelf pop out so that the CD can be placed in it. Pressing the button again causes the shelf to retract into the drive so that it can play the CD. This type of drive is referred to as a caddieless drive. Both a caddie drive and a caddieless drive are depicted in Figure 12.12.

Figure 12.12

Caddie and caddieless CD-ROM drive units.



CD-ROM Interfaces

With as many speed choices as there are, there are also the following three choices of architectures for CD-ROM drives:

- SCSI
- IDE
- Proprietary interfaces

The SCSI and IDE drives can use a controller already installed in the computer. However, there are many versions of the SCSI standard, which has not necessarily been standardized. There are also two versions of the IDE interface: the original IDE and a newer, enhanced IDE, or EIDE. The EIDE interface has been redefined to allow faster transfer rates, as well as the handling of more storage capacity. It can also be used to control drive units such as a tape or CD-ROM. The EIDE interface is often described as an ATAPI (AT Attachment Peripheral Interconnect), or a Fast ATA (Fast AT Attachment) interface.

The controllers for proprietary interfaces are often included with the drive or with a sound card. To gain full advantage of a CD-ROM, it is becoming essential to also have a sound card. Many sound cards include a CD-ROM drive interface or controller. However, these controllers may not be IDE or SCSI compatible. They often contain proprietary interfaces that only work with a few CD-ROM models.

Software drivers are required to operate a CD-ROM drive from DOS or Windows. Normally, the drivers come packaged with the drive if an IDE interface is being used. Conversely, the drivers are included with the interface card when a SCSI interface is employed.

CD Writers

Objective

Another type of CD drive is classified as `Write Once, Read Many` (WORM) drive. As the acronym implies, these drives enable users to write information to the disk once, and then retrieve this information as you would with a CD-ROM drive. With WORM drives, after information is stored on a disk, the data cannot be changed or deleted.

Typically, CD-ROM drives and disks conform to the International Standards Organization's ISO-9660 specification. This standard is an advancement of an earlier pseudo CD standard called the High Sierra format. The presence of established formats have allowed CD writing technologies to evolve into a form suitable for use in personal computers.

CD Writer technology has developed to the point where it is inexpensive enough to be added to a typical PC system. CD Writers record data on blank CD-Recordable (CD-R) disk. A CD-R is a Write Once, Read Many (WORM) media that is generally available in 120 mm and 80 mm sizes. The drives are constructed using a typical 5.25-inch, half-height drive form factor. This allows it to conveniently fit into a typical PC drive bay.

Although CD-Rs resemble CD-ROM disks, they are not exactly the same. The construction of the CD-R disk is considerably different than the CD-ROM disk. The writable disk is constructed as described in Figure 12.13. The CD-R disk is created by coating a transparent polycarbonate substrate with an opaque dye. The dye is then covered with a thin layer of gold and topped with a protective lacquer layer and a label. The WORM writing mechanism is not as strong as that of a commercial CD-ROM duplicator. Instead of burning pits into the substrate of the disk, the CD Writer drive uses a lower-powered laser to discolor the dye material. Therefore the WORM disk is difficult to copy.

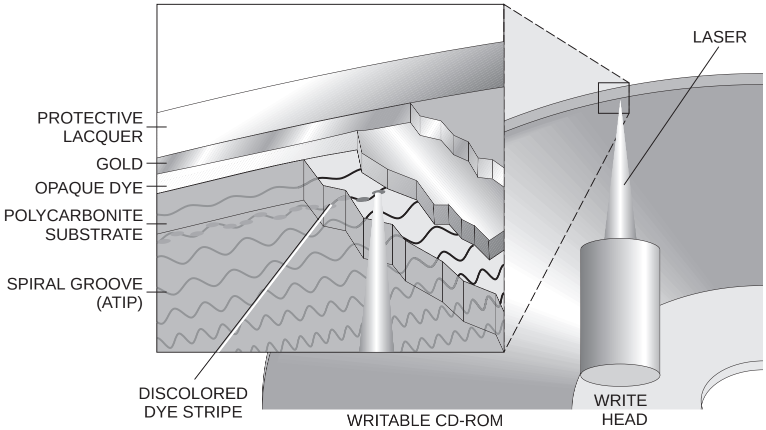


Figure 12.13

Writable CD-R disk.

In other respects, the WORM disk format is identical to that of a CD-ROM, and information written on it can be read by a typical CD-ROM drive. The spiral track formation and sectoring are the same as those used with CD-ROM disks. In addition, the CD Writer can produce recordings in the standard CD book formats (that is, Red, Yellow, Orange, and Green).

During the write operation, the CD Writer encodes the data into the designated CD storage format. In this process, the Writer adds error-correction code information and provides data interleaving. The Writer then uses a high-intensity laser to write the data on the thermally sensitive media. The laser light is applied to the bottom side of the disk. It passes through the substrate to the reflective layer and is reflected back through the substrate. The light continues through the drive's optics system until it reaches the laser detector.

Unlike the pits and lands arrangement of the CD-ROM disk, the CD-R uses a polymer dye, such as cyanine or pthalocyanine, as the recording medium. The polymer held in a shallow pregroove molded into the substrate material. This groove is used by the drive's positioning circuitry to maintain track alignment. The drive also uses a built-in wobble in the groove for rotational speed control of the disk. The wobble is created so that it delivers a 22.05 KHz signal when rotated at the correct velocity. This signal is also encoded with Absolute Time In Pre-groove (ATIP) information that the system uses to keep track of information stored on the disk.

When the polymer is exposed to the light of the high-intensity writing laser, it heats up and becomes transparent. This exposes the reflective gold layer beneath the polymer. During read back, the reflective layer reflects more light than the polymer material does. The transitions between lighter and darker areas of the disk are used to encode the data.

The data can be written to the CD-R using one of three methods:

- As a disk-at-once
- As a session-at-once
- As a track-at-once

A `session` is the CD equivalent of an HDD partition. The ISO 9660 standard requires that each session contain a Lead In section, a Program area, and a Lead Out section. The Lead In section contains the table of contents for the session (there is no FAT on a CD-ROM or WORM disk). The Lead Out section holds information that marks the end of the program area.

Most CD Writers are packaged with a SCSI interface and host adapter. This interface must supply the stream of data to the disk at an acceptable rate for encoding and recording on the CD-R. CD Writers are typically able to write to the CD-R at either 1x or 2x CD speeds. These settings have nothing to do with playback speeds.

Installing a CD-ROM Drive

The installation process for a CD-ROM drive is identical to that of a hard or floppy drive. For an internal drive, confirm the drive's Master/Slave/Single or SCSI ID configuration setting, install the CD-ROM unit in one of the drive bays, connect the power cord and signal cable, and load the driver software. External CD-ROM drives are even easier to install. Just connect the power, run the signal cable between the drive unit and the system, confirm the Master/Slave/Single or SCSI ID setting, and install the driver software.

Internal Units

Figure 12.14 illustrates the installation of an internal CD-ROM drive. If the interface type is different than that of the HDD, it will be necessary to install a controller card in an expansion slot. To accomplish this, remove the cover from the system unit. To install a controller, locate a compatible, empty expansion slot in the system unit. Remove the expansion slot cover and install the internal controller into the expansion slot. Replace the screw to secure the controller card to the system unit's back panel. Finally, refer to the Owner's Manual regarding any necessary jumper or switch settings.

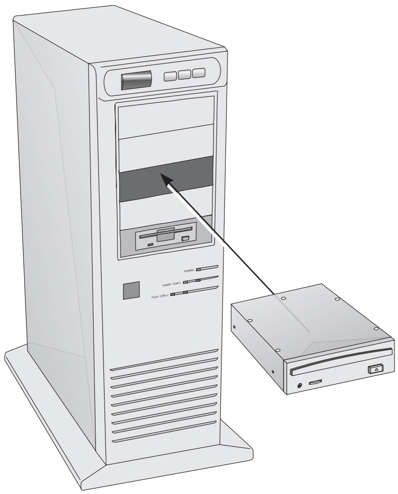


Figure 12.14

Installing an internal CD-ROM drive.

Install the drive in the system by sliding the drive into its bay (near the HDD unit if possible), and securing it with two screws on each side.

To connect the drive to the system, connect the CD-ROM drive to the HDD signal cable, observing proper orientation (unless a sound card with a built-in controller is being used). Verify that the other end of the cable is connected to the CD-ROM drive. Connect one of the power supply's options connectors to the CD-ROM drive. Connect the audio cable to the drive and to the sound card's CD Input connection (if a sound card is installed). Finish the job by replacing the system unit's cover and booting up the system.

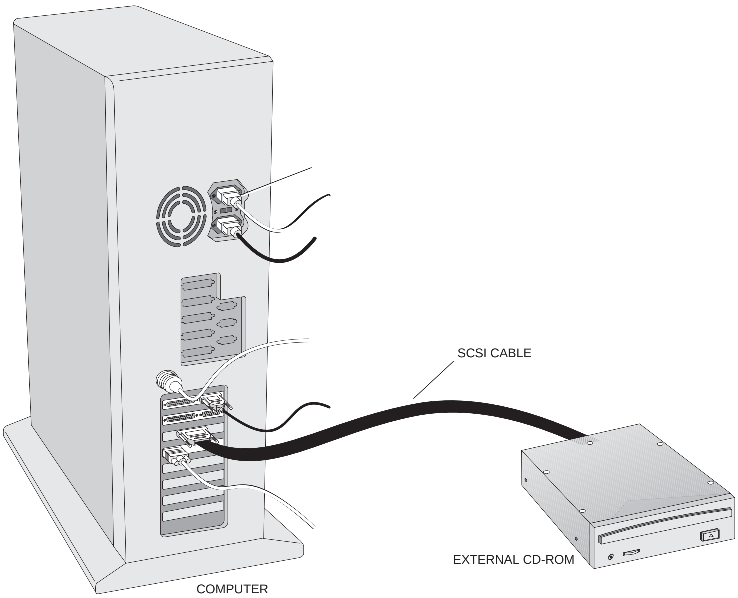
External Units

Because of the popularity of notebook computers, some manufacturers have produced external CD-ROM drives. These drives typically connect to a SCSI host adapter or to a parallel port. This requires a fully functional, bidirectional parallel port and a special software device driver. Some internal drives include driver software. Beginning with DOS version 6.2, however, a driver called MSCDEX is included to identify and assign a drive letter to the CD-ROM.

Figure 12.15 illustrates the installation of an external SCSI CD-ROM drive. Because the drive is external, connecting the CD-ROM unit to the system usually only involves connecting a couple of cables together. First, connect the CD-ROM's power supply to the external drive unit. Before making this connection, verify that the power switch or power supply is turned off. Connect the signal cable to the computer. Finally, connect the opposite end of the cable to the external CD-ROM unit. Complete the installation by installing the CD-ROM driver software on the system's hard-disk drive.

Figure 12.15

Installing an external CD-ROM drive.



Configuring a CD-ROM Drive

As previously indicated, the CD-ROM Drive unit must be properly configured for the system it is being installed in. In an IDE system, the Master/Slave setting must be confirmed. In a SCSI system, the ID setting must be correct. In a SCSI system, the only requirement is that a valid ID setting is configured. However, in an IDE system, some thought may be required as to how to configure the drive.

In a single HDD system, the CD-ROM drive is normally set up as the slave drive on the primary interface. However, in a two-HDD system, the CD-ROM drive would most likely be configured as the Master or Single drive on the secondary interface. If the system also contains a sound card with a built-in IDE interface, it should be disabled to prevent it from interfering with the primary or secondary interfaces.

After the CD-ROM's hardware has been installed, it will be necessary to install its software drivers. Consult the owner's manual for instructions on software installation. Typically, all that is required is to insert the OEM driver disk in a floppy drive, and follow the Manufacturer's directions for installing the drivers. If the drive fails to operate at this point, reboot the system by using single-step verification and check the information on the various boot screens for error messages associated with the CD-ROM drive.Driver signatures are generally written in the form of `MSCD000`. To do this, open the Write program from the Windows Accessories window, check the AUTOEXEC.BAT file for a line that says: `C:\XXX\MSCDEX`, and then check the CONFIG.SYS file for a line that says `Device=C:\XXX\XXX /D:MSCDXXX`. The `XXX`s will be replaced by the manufacturer's parameters. You can also examine the SYSTEM.INI file for the CD-ROM device statement.

In Windows 95, an advanced CD-ROM driver called CDFS (CD-ROM File System) has been implemented to provide protected-mode operation of the drive. Win95 retains the MSCDEX files for real-mode operation. If Win95 detects that the CDFS has taken over control of the CD-ROM on its initial bootup, it will REM the MSCDEX lines in the AUTOEXEC.BAT file.

Troubleshooting a CD-ROM Drive

The troubleshooting steps for a CD-ROM drive are almost identical to those of an HDD system. The connections and data paths are very similar. Basically, four levels of troubleshooting apply to CD-ROM problems. These are the configuration level, the DOS level, the Windows level, and the hardware level. Figure 12.16 shows the parts and drivers associated with CD-ROMs.

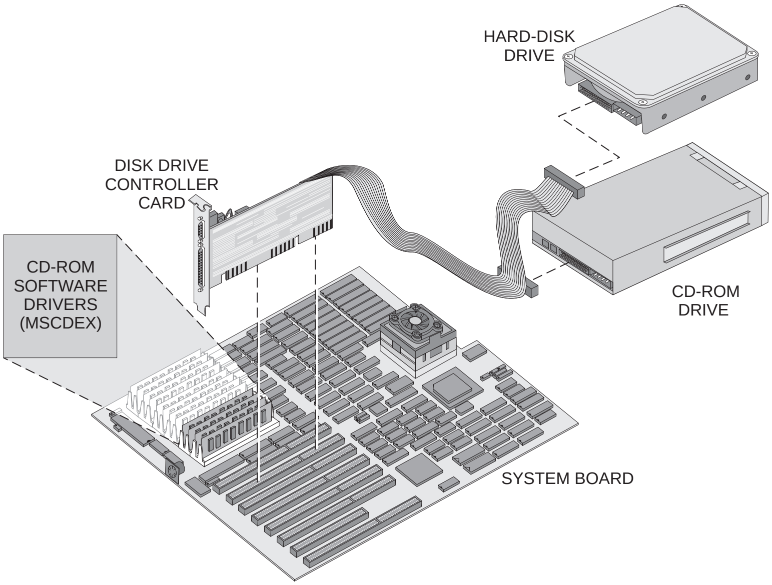


Figure 12.16

Components and drivers associated with CD-ROMs.

CD-ROM Software Checks

Many of the software diagnostic packages available include test functions for CD-ROM drives. Try to choose a diagnostic package that includes a good variety of CD-ROM test functions. Run the program's CD-ROM function from the Multimedia selection menu. Select the equivalent of the program's Run All Tests option.

Use a diagnostic program such as MSD to check the IRQ and I/O address settings for possible conflicts with other devices. If the settings are different than those established by the hardware jumpers on the controller, change the settings so that they both match each other and do not conflict with other devices.

Windows 3.x Checks

Begin by examining the SYSTEM.INI file for the CD-ROM device statement. This is accomplished by typing **Edit SYSTEM.INI** at the DOS prompt and pressing the Enter key. Inside the file, locate the CD-ROM driver by paging down to locate the `DEVICE=MSCDEX` line. If the line is present, check the CD-ROM documentation for troubleshooting hints. If the line is missing, reinstall the driver software. Check for the MSCDEX statement in the AUTOEXEC.BAT file.

Add a **LANMAN10.DOS** line to the SYSTEM.INI file. At the DOS prompt type **Edit SYSTEM.INI**. Inside the file, locate the `[386enh]` section heading, and add the line **DEVICE=LANMAN10.DOS** to the list of device statements. Check the Windows System subdirectory for the LANMAN10.DOS file.

If the correct driver is not installed, load the correct driver or contact the CD-ROM manufacturer for the correct Windows driver.

Windows 95 Checks

In Windows 95, the CD-ROM can be accessed through the CD icon in the desktop's My Computer icon. The CD-ROM drive's information is contained in the Control Panel's System icon. The properties of the installed drive are located under the Settings tab. Figure 12.17 shows a typical set of CD-ROM specifications in Windows 95.

Check the system's AUTOEXEC.BAT and CONFIG.SYS FILES for updated information. To do this, open the WordPad program from the Win95 Accessories listing, check the AUTOEXEC.BAT file for a line that says `REM C:\XXX\MSCDEX`, and then check the CONFIG.SYS file for a line that says `REM Device=C:\XXX\XXX /D:MSCDXXX`. If the correct drivers are not installed, load the correct driver or contact the CD-ROM manufacturer for the correct Windows driver.

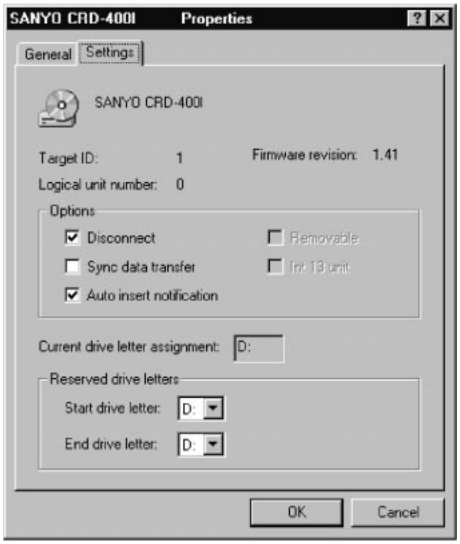


Figure 12.17

Control Panel/ Device Manager/ Settings.

CD-ROM Hardware Checks

If the configuration and software checks do not remedy the CD-ROM problem, it will be necessary to troubleshoot the CD-ROM–related hardware. Basically, the hardware consists of the CD-ROM drive, the signal cable, the power cord, and the controller. The controller may be mounted on a MI/O card or, in a Pentium system, on the system board. For external drives, the plug-in power adapter will need to be checked as well.

In most systems, the CD-ROM drive shares a controller with the hard-disk drive. Therefore if the hard drive is working and the CD-ROM drive is not, the likelihood that the problem is in the CD-ROM drive is very high.

Before entering the system unit, check for simple user problems. Is there a CD-ROM disk in the drive? Is the label side of the disk facing upward? Is the disk a CD-ROM, or some other type of CD?

If no simple reasons for the problem are apparent, begin by exchanging the CD-ROM drive with a known good one of the same type. For external units, just disconnect the drive from the power cord and signal cable, and substitute the new drive for it. With internal units, it will be necessary to remove the system unit's outer cover and disconnect the signal cable and power cord from the drive. Remove the screws that secure the drive in the drive bay. Install the replacement unit and attempt to access it.

If the new drive works, reinstall any options removed and replace the system unit's outer cover. Return the system to full service, and service the defective CD-ROM drive accordingly.

If the drive will not work, check the CD-ROM drive's signal cable for proper connection at both ends. Exchange the signal cable for a known good one.

If the drive still refuses to operate, turn the system off and exchange the controller card (if present) with a known good one. In Pentium systems, the controller is mounted on the system board. In other systems, the controller is normally mounted on an MI/O card. Disconnect the disk drive's signal cable from the controller card and swap the card with a known good one of the same type. If the controller is built into the system board, it may be easier to test the drive and signal cable in another machine than to remove the system board. Make certain to mark the cable and its connection points to ensure proper reconnection after the exchange.

Reconnect the signal cable to the controller and try to reboot the system with the new controller card installed. If the system boots up, reinstall any options removed and replace the system unit's outer cover. Return the system to full service, and service the defective controller card as appropriate.

If the drive still refuses to work, check to see whether the CD-ROM drive has been properly terminated. Exchange the CD-ROM's power connector with another one to make sure that it is not a cause of problems. Finally, exchange the system board with a known good one.

Sound Cards

Objective

The sound-producing capabilities of early PCs was practically non-existent. They included a single, small speaker that was used to produce beep-coded error messages. Even though programs could be written to produce a wide array of sounds from this speaker, the quality of the sound was never any better than the limitations imposed by its small size.

Audio is an important part of the multimedia world. People typically think of audio in computer systems as voice communication. This is normal, because most of the early work in computer audio applications was performed in the areas of speech synthesis and recognition. This work was fueled by the desire to produce human/computer interfaces that would appeal to the public. If people could communicate with the computer, and it could talk to them, there would be no need for complicated programming, and the computer would be a handy tool for the masses.

The more recent advancements made in the area of multimedia have redirected this research into more computer/TV/radio applications. Voice and music are an integral part of every movie and advertising presentation you hear. People make their living just by creating short musical presentations (called jingles) that will catch the interest of the target audience.

A typical audio digitizer system is depicted in Figure 12.18. A microphone converts sound waves from the air into an encoded, analog electrical signal. The analog signal is applied to the audio input of the sound card. On the card, the signal is applied to an A/D converter circuit, which changes the signal into corresponding digital values, as described in Figure 12.19. The sound card takes samples of the analog waveform at predetermined intervals and converts them into corresponding digital values. Thus the digital values approximate the instantaneous values of the sound wave.

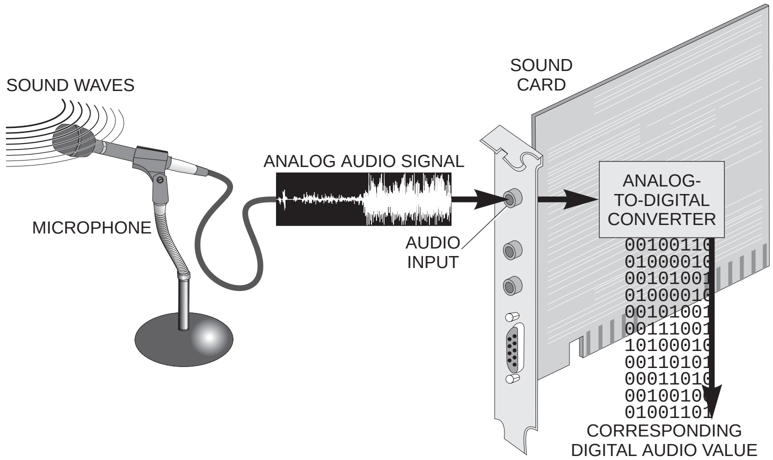


Figure 12.18

A typical audio digitizer system.

The `fidelity` (the measure of how closely the original sound can be reproduced) of the digital samples depends on two factors: the accuracy of the samples taken, and the rate at which the samples are taken. The accuracy of the sample is determined by the `resolution` capabilities of the A/D converter. Resolution is the capability to differentiate between values. If the value of the analog waveform is 15.55 micro-volts at a given point, how close can that value be approximated with a digital value? Resolution of an A/D converter is determined by the number of digital output bits it can produce. An 8-bit A/D converter can represent up to 256 (2^8) different values, for example. On the other hand, a 16-bit A/D converter can represent up to 65,536 (2^{16}) different sound levels. The more often the samples are taken, the more accurately the original waveform can be reproduced.

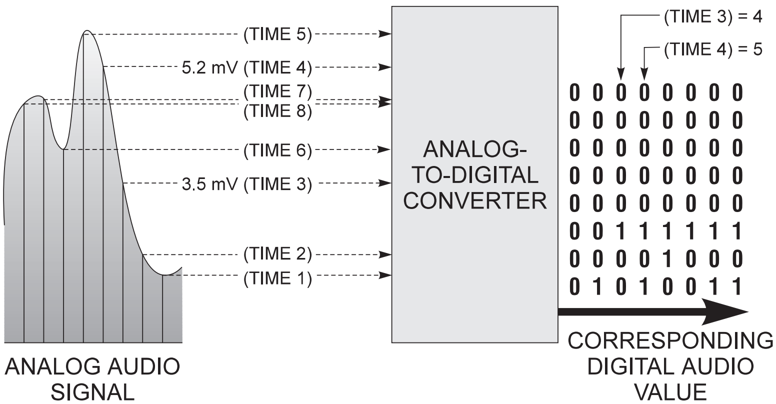


Figure 12.19

Converting signal changes to digital values.

Playback of the digitized audio signal is accomplished by applying the digital signals to a D/A converter at the same rate the samples were taken. When the audio files are called for by the application software, the Runtime Module of the authoring software sends commands to the audio output controller on the sound card. The digitized samples are applied to the audio output IC and converted back into the analog signal. The analog signal is applied to an audio preamplifier that boosts the power of the signal and sends it to an RCA or mini jack. This signal is still too weak to drive conventional speakers. However, it can be applied to an additional amplifier or to a set of speakers that have an additional amplifier built into them.

A CD-quality audio signal requires a minimum of 16-bit samples, taken at approximately 44 KHz. If you calculate the disk space required to store all the 16-bit samples collected in one minute of audio at this rate, the major consideration factor associated with using digitized audio becomes clear ($16 \times 44{,}000 \times 60 \times 8 = 5.28$ MB). If you want stereo sound, this will double to a whopping 10.56MB. Therefore, CD-quality audio is not commonly used in multimedia productions. The audio sampling rate used in multimedia titles is generally determined by the producer. Another alternative is to limit the digitized audio used in a product to short clips.

Installation

Installing a sound card is similar to installing any other adapter card. Refer to the card's User's Guide to determine what hardware configuration settings may need to be made before inserting the card into the system. It may also be beneficial to run a diagnostic software package to check the system's available resources before configuring the card.

After the hardware configuration is complete, just install the card in one of the system's vacant expansion slots and secure it to the back panel of the system unit. Plug the microphone and speakers into the proper jacks on the card's back plate. With the card installed in the system, load its software drivers according to the directions in the User's Guide.

Figure 12.20 depicts the connection scheme of a typical sound card. Most sound cards support microphones through stereo RCA jacks. A very similar speaker jack is also normally present on the back of the card. Depending on the card, the jack may be designed for mono or stereo output. An onboard volume control wheel may also protrude through the card's back plate.

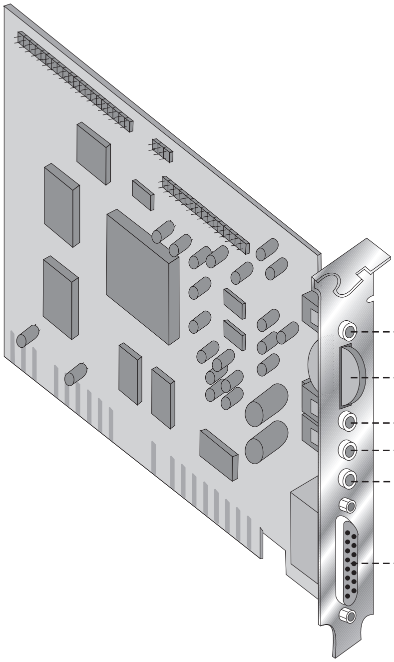


Figure 12.20

Back-panel connections of a sound card.

A typical Sound Blaster–compatible I/O address range used by sound cards is 220_h through $22F_h$. Alternative address ranges include 230–$23F_h$, 240–$24F_h$, and 250–$25F_h$. Typically, the sound card will use interrupt request channel 7 as the default. DMA channel 1 is used as well.

Some sound cards contain game port adapter circuitry for attaching joysticks to the sound card. The game port circuitry must conform to the (201_h) I/O addressing specifications, described in Chapter 5, to remain compatible with the IBM standard. Advanced sound cards may also include a MIDI port connection.

Audio Software

When the Run Module of a multimedia title encounters a request for an audio file, it locates the file requested and applies it to the sound card's audio driver software. The driver software sets up the sound card to produce the sound file. The sound card has two ways to produce the requested sound file. It can retrieve a prerecorded file from memory or hard disk and replay it, or it can synthesize it.

A `synthesized` sound file is one that is created, or generated, at the time it is called for. Two popular synthesis techniques are in use. These are FM synthesis and wave table synthesis. FM, or frequency modulated synthesis, creates sounds by generating programmed combinations of various sine wave frequencies to imitate the desired sound. Most of the sound synthesizers used with PCs are FM synthesizers. Although the actual synthesizer is an IC component, the creation of the sound is controlled by the audio software.

Wave table synthesis is a high-quality sound-generating method used to imitate the sounds made by real musical instruments. The table contains small sound clips of each of the natural instruments to be synthesized. The synthesizer processes the small clips to produce all the sounds called for by the software. The small clips for the instruments are generally stored in a compressed format, in a ROM chip on the adapter card.

The sound card's audio software may also include a user software interface, like the one depicted in Figure 12.21. This particular interface is designed so that it resembles the control panels associated with a CD-player, a MIDI player, a tape player, and a graphic equalizer.

The software buttons in the interface enable the user to control the system's audio capabilities. The unit can play audio CDs, WAV files, and MIDI files.

Figure 12.21

A user interface.

MIDI

Objective

Most sound cards possess only the capability to capture audio signals, digitize them, and play them back as they were recorded. Some sound cards have the capability to generate synthetic sounds that are not a function of a digitizing process. The MIDI standard was created by musical instrument makers to allow music synthesizers and other electronic music devices to communicate with computers and with each other.

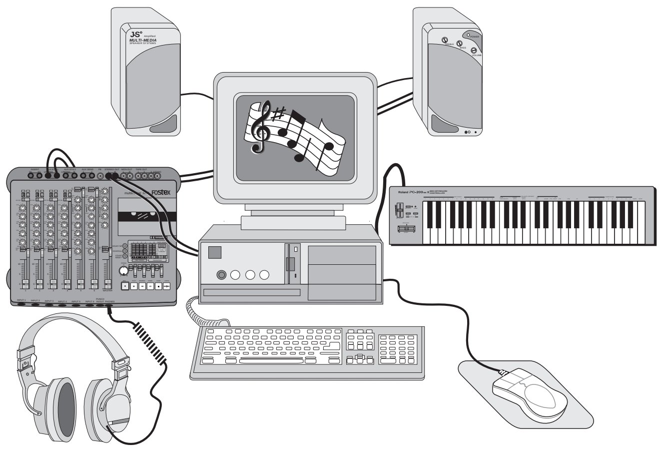
The MIDI specification began as a hardware-connectivity format. It included a protocol for exchanging data and a cabling scheme for hooking devices together. The agreement was so widely accepted by the music industry that virtually every electronic instrument manufactured today conforms to the MIDI standard.

The MIDI interface protocol lists 16 instrument channels that can be used. Commands can be sent on all channels to control a number of voices. The standard defines 128 MIDI program codes called the General MIDI Standard. These codes ensure that the sound produced by different MIDI instruments will be consistent.

Figure 12.22 shows a typical MIDI system. The system contains a MIDI-equipped computer, a keyboard controller/synthesizer, an audio mixer/recorder, and related sound modules. The computer contains a MIDI interface card. Although a mixer has been shown in this figure, advances in MIDI software have led to systems where the mixer has been eliminated in favor of software mixing. Newer MIDI software contains programming called MIDI Machine Control (MMC). Under MMC, the software actually controls the intelligent MIDI devices such as mixers, stage lights, and so on.

Figure 12.22

A typical MIDI system.



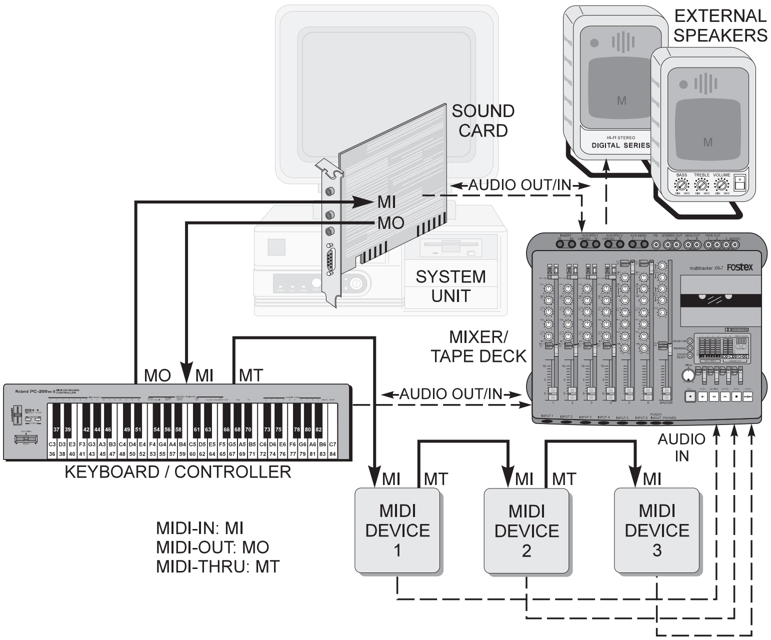
Sophisticated MIDI systems, with a large number of instruments, will still opt for a hardware mixing console. A sound module is actually a hardware component containing ROM devices that hold the sampled sounds of the real instruments being produced.

All MIDI devices communicate serially through round, 5-pin DIN connectors, as described in Figure 12.23. Three types of connections are possible in a MIDI system. These are the MIDI-In, MIDI-Out, and MIDI-Thru connections. A single connection cable can be used for all three connection types. The synthesizer/controller requires two connections to the MIDI interface in the computer. The first deals with the controller portion of the keyboard. A MIDI cable runs from MIDI-Out of the controller to MIDI-In of the interface. On the synthesizer side of the keyboard, a MIDI-In from the keyboard must be connected to MIDI-Out of the interface card.

To continue the MIDI connection scheme, the interface would require an additional MIDI-Out connection. Alternately, MIDI-Thru connections can be used to serially connect all the other MIDI devices to the system, as described in Figure 12.24.

Figure 12.23

MIDI cable connections.



The various devices are connected to the mixer/recorder through audio out/in patch cords.

MIDI data transfers are conducted serially. Each MIDI device contains a MIDI controller, as does the MIDI adapter card in the computer system. In the MIDI device, the data produced by the equipment is applied to the MIDI controller, which converts the data into the MIDI data format. The signal passes serially to the MIDI adapter card in the computer. After processing, the computer sends it back to the MIDI device.

The MIDI data stream can contain a large amount of information about the instrument. The information in general MIDI data includes the patch (instrument), the MIDI channel (1 of 16), the note played, and velocity setting (0–127). The patch code is also referred to as the program change, or PC. The MIDI note being played is actually controlled by two separate transmissions. The Note-on code starts the note to be played, while a separate, Note-off code is used to end the note being played. Velocity is a setting for the loudness of the note being played. The data stream can also contain instrument-specific information. This type of transfer is referred to as system exclusive (SYSX) data. In this format, all other instruments will ignore the data not directed at them.

Speakers

Typically, the output of the sound card requires additional amplification if external speakers are used. The amplification circuitry is normally included in the external speaker units. Power for these speakers is derived from batteries housed in the speaker cabinets or from a small, AC power converter. Most sound cards do have the capability to directly drive low-power headphones. Audio output can also be produced by the system's internal speaker, or it can be amplified through external audio amplifier systems for applications such as surround sound. Figure 12.24 depicts various speaker arrangements used with multimedia systems.

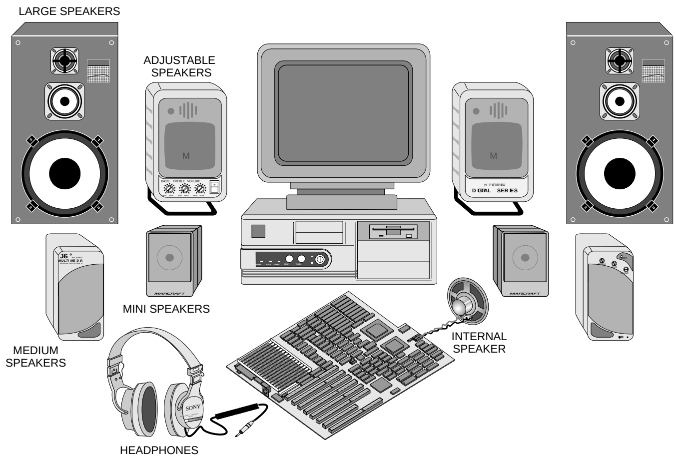


Figure 12.24

Typical speaker arrangements.

Troubleshooting Sound Cards

The components involved in the audio output of most computer systems is very simple. There is a sound card adapter, some speakers, the audio-related software, and the host computer system. Fortunately, there are diagnostic software packages capable of testing sound card operation.

Most sound cards perform two separate functions. The first is to play sound files, and the second is to record them. It may be necessary to troubleshoot problems for either function.

Sound Card Configuration Checks

If sound problems are occurring in the multimedia system, two of the first things to check are the hardware and audio software configuration settings. Refer to the sound card manufacturer's documentation for proper hardware settings. These items usually include checking the card's jumper settings for IRQ and I/O address settings. With more Plug and Play cards on the market, however, software configuration of IRQ and I/O addressing is becoming more common.

In the past, sound cards have been notorious for interrupt conflict problems with other devices. Because these conflicts typically exist between peripheral devices, they may not appear during bootup. As an example, if the sound card operates correctly except when a printing operation is in progress, an IRQ conflict probably exists between the sound card and the printer port. Similar symptoms would be produced for tape backup operations if the tape drive and the sound card were configured to use the same IRQ channel. Use a diagnostic program, such as MSD, to check the system for interrupt conflicts. Use a diagnostic program, such as MSD, to check the system for interrupt conflicts.

Sound Card Software Checks

Many diagnostic packages offer testing features for sound cards and other multimedia-related components. Sound card problems should not prevent the system from loading a software diagnostic, so run the All Tests equivalent in the Multimedia section of the diagnostic package. Also, run checks to see whether addressing (IRQ or DMA) conflicts are causing a problem. If so, reconfigure the system's components so that the conflicts are removed.

Is the software application running a DOS version? If so, it may not be able to output audio under Windows. In enhanced mode, Windows will not hand over control to the DOS application. In such cases, a message saying `This application will not be able to use audio` should appear. Run the application from DOS to see whether it runs, or start Windows in standard mode by typing **WIN /S** at the DOS prompt, before pressing the Enter key.

The Sound Recorder can be used to check the operation of WAV files under Windows. If the audio file will not play from the Sound Recorder, make sure the Sound Recorder is working by attempting to play audio files that have played on the system before. If the files play from the Sound Recorder, the other application being used to try to play the file needs to be examined for proper installation and setup.

If the Sound Recorder will not play audio files through the sound card, check to see that the multimedia icons are installed in the Control Panel and in Accessories windows. Also check the Control Panel's Drivers icon to see that the correct audio driver is installed, and that its settings match those called for by the sound card manufacturer. If the icons are missing, add the icons through the Setup function in the Main window.

If the driver is not installed or is incorrect, add the correct driver from the available drivers list. If the correct driver is not available, reinstall it from the card's OEM disk or obtain it from the card's manufacturer.

If the Windows Media Player won't play MIDI files, look in the Drivers section of the Control Panel to see that the MIDI driver is set up properly. Set the Media Player to play MIDI files, and make certain that the MIDI mapping is correct. Check to see whether more than one MIDI device is connected. If so, disconnect the other MIDI devices. Also, set the MIDI device to Mode 1 for testing purposes. Mode 1 is the most flexible, and allows even distorted sounds to be played.

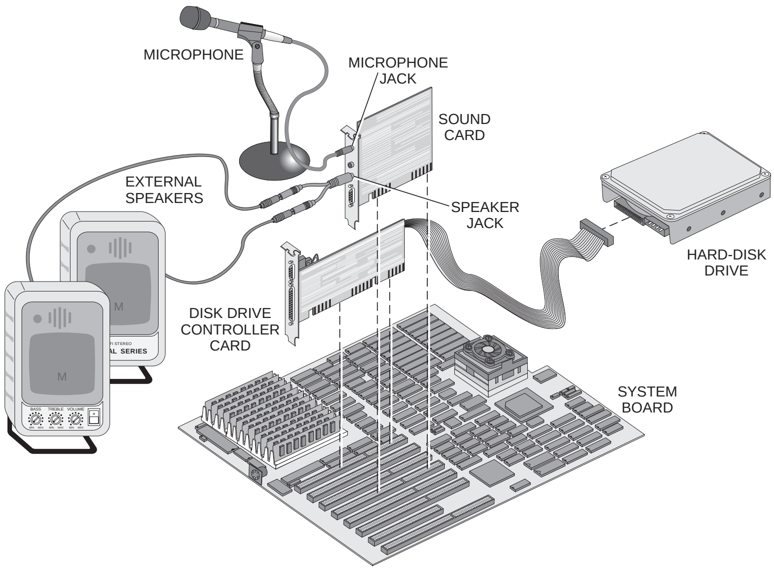
If audio files will not play through the Media Player, check to see that the file is a WAV file. Check in the Device menu to see that the Media player is set up to play audio files. Check the audio driver installation to see that the proper audio driver is installed.

Sound Card Hardware Checks

Provided that the sound card's configuration is properly set, and the software configuration matches it, the sound card and speakers will need to be checked out. The system's sound card-related components are described in Figure 12.25. Most of these checks are very simple. They include checking to see that the speakers are plugged into the speaker port. It is not uncommon for the speakers to be mistakenly plugged into the card's MIC (microphone) port. Likewise, if the sound card will not record sound, make certain that the microphone is installed in the proper jack (not the speaker jack), and that it is turned on. Check the amount of disk space on the drive to ensure that there is enough to hold the file being produced.

Figure 12.25

Sound card-related components.



If the system will not produce sound, troubleshoot the audio output portion of the system. Do the speakers require an external power supply? If so, is it connected, and are the speakers turned on? If the speakers use batteries for their power source, check them to see that they are installed and good. Check the speakers' volume setting to make certain they are not turned down.

Video Capture Cards

Objective

Video capture cards are responsible for converting video signals from different sources into digital signals that can be manipulated by the computer. Like the audio conversion process, the video card samples the incoming video signal by feeding it through an A-to-D converter, as depicted in Figure 12.26.

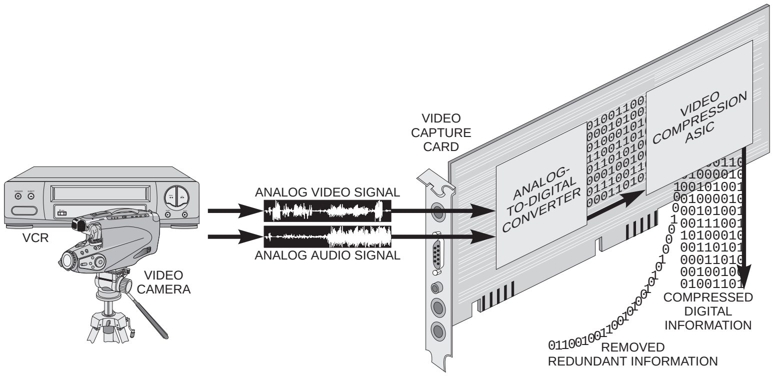


Figure 12.26

Converting an incoming video signal.

The digitized output from the A-to-D converter is applied to a video compression ASIC. The compression chip reduces the size of the file by removing redundant information from consecutive frames. This reduction is necessary because of the extreme size of typical, digitized video files. Video compression schemes can reduce the size of a video file by a ratio of up to 200:1.

As the sections of video are compressed, the reduced files may be applied to the system's RAM memory, or routed directly to the hard-disk drive. The audio signal is not compressed, but it is synchronized to the video signal so that it will play in the right places when the video is rerun.

When the digitized video is recalled for output purposes, the file is reapplied to the compression chip, which restores the redundant information to the frames. The output from the compression chip is applied to the digital-to-analog portion of the video-processing circuitry. The analog signals are converted back into the proper VGA format and applied to the video-out connector at the back plate of the card.

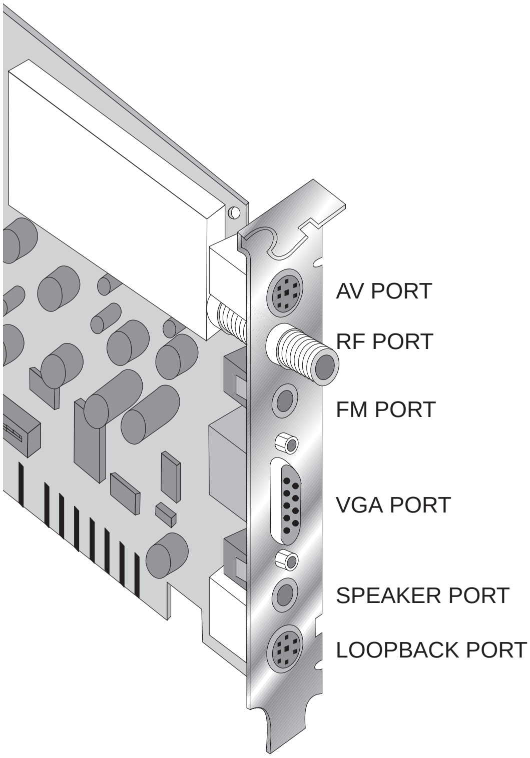
Depending on the compression methods used, the video clip can be played back from the Windows AVI structure or through the capture card. This is a major consideration when creating a title that includes a video component. Will the user's computer have a compatible video digitizer card installed, or should it be expected that the video clip will play through the Windows multimedia extensions?

Capture Card Signals and Connections

The connection points for a typical video capture card are displayed in Figure 12.27. Sources for video capture normally include VCRs and camcorders. Some capture cards, like the one shown in Figure 12.27, include an RF demodulator and TV tuner so that video can be captured from a television broadcast signal or a cable TV input.

Figure 12.27

Connections of a typical video capture card.



The output from these video-producing devices tends to be composite TV, or analog S-video signals. A video decoder circuit is used to convert the analog signal into a stream of digital signals. However, these are not the RGB digital signals useful to the VGA card. The characteristics of the decoded TV signal are defined in television industry terms as YUV. The Y portion of the term refers to the luminance of the signal color; the UV portion describes the color component of the signal.

One of the jobs of the video capture card is to convert the YUV format into an RGB VGA-compatible signal. An encoding circuit samples the incoming analog signal, and then performs an operation known as `color space conversion` on it. Color space conversion is the process of converting the YUV signal into the RGB format acceptable to the VGA card's screen memory. The resolution of a studio-quality TV signal is defined as 512×512 pixels, delivered in two interlaced screens, at a rate of 60 per second. The encoder converts this signal scheme into a 640×480 (in VGA mode) image delivered to the screen in a single, noninterlaced screen, at a rate of 30 per second.

In addition to changing the format, the capture card also scales the image to fit in the defined video window on the monitor's screen. The capture card's video signal processor adjusts the image to the correct size, by interpolating (adding or removing) adjacent pixels as necessary.

The encoder samples the analog signal at a rate of 27MB per second. This value becomes very important when you realize that, at this rate, a 500MB hard drive would be full in 18.5 seconds. To make the digitized video manageable and useful to the digital computer system, the signal must be compressed into smaller files.

Compression Standards

Objective

To date, two common data compression standards are employed with digitized video. These are the Joint Photographic Experts Group (JPEG) and the Moving Picture Experts Group (MPEG) compression standards.

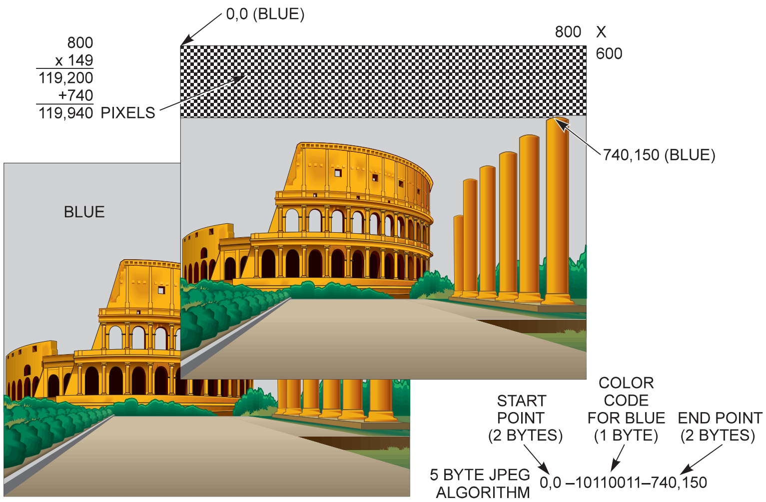
JPEG

For still images, the JPEG compression standard is used. JPEG compression also removes redundant picture information from still video frames. JPEG compresses still frames in 1/30 of a second. Its playback capabilities allow it to also reproduce frames at a rate of 30 per second. This allows JPEG to play back digitized movies at full-screen sizes. Used in this manner, the format is called M-JPEG. However, the M-JPEG format has not been standardized. It is therefore vendor specific at this point.

Most compression algorithms use complex mathematical formulas to remove redundant bits, such as successive 0's or 1's, from the data stream. When the modified stream is played back through the decompression circuitry, the formula reinserts the missing bits to return the data to its original state. This concept is depicted in Figure 12.28.

Figure 12.28

Examining successive bytes in a frame.



Ideally, it would be assumed that a compressed/decompressed image would be returned to 100% of its original state. In practice, however, some information from the original image will not be completely restored. In a JPEG system, the fidelity of the image depends on the compression ratio used. JPEG can obtain compression ratios of 100:1, however, the quality of the reproduced image may be quite poor. As a matter of fact, when the compression ratio exceeds 20:1, the decompressed image deteriorates noticeably.

The fidelity of the JPEG system depends on the complexity of its compression algorithm. Some expensive compression systems are lossless JPEG systems. Their returned image quality is very good, but the compression ratio is typically very low. Other JPEG systems provide better compression, but at the cost of image loss. For this reason, most JPEG capture is performed at less than full-screen size, or with less than every frame. Reduced movie sizes are shown in 1/4-screen windows. Reduced frame rates are 1/2 (15 fps), or 1/3 (10 fps).

MPEG

Although JPEG provided enough compression to allow single-frame digitized images to fit on disk drives, it soon became apparent that full-motion pictures were going to need much greater compression to be useful on current technology. Therefore the MPEG format delivered a compression algorithm that provides compression ratios up to 200:1, with high-quality video and audio.

Like JPEG, MPEG removes redundant picture information from individual scenes. Instead of just removing redundant information from within a single frame, however, the MPEG compression scheme removes redundant information from consecutive scenes. In addition, the MPEG methodology only compresses key objects within a frame every 15th frame. Between these key frames, only the information that changes from frame to frame is recorded. In this manner, MPEG compression systems can attain the compression ratios and good image fidelity required by full-motion video.

MPEG files are difficult to edit because the content of each frame relates heavily to the contents of the frames around it. The MPEG recording scheme is illustrated in Figure 12.29.

The MPEG standard includes specifications for audio compression and decompression, in both MPEG 1 and 2. MPEG 1 supports a very near CD-quality stereo output, at data rates between 128 and 256Kbps. The MPEG 2 specification supports CD-quality surround-sound (4-channel) output.

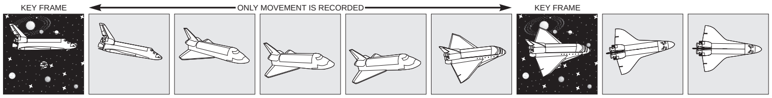


Figure 12.29

The MPEG recording scheme.

Other Compression Standards

Another data compression method used with PCs is the Indeo compression standard, developed by Intel. It is similar to the MPEG standard, in that it was actually designed to be a distribution format. It was primarily intended to play back compressed video files from the smallest file size possible. Later versions of this standard include the MPEG compression methods.

Another compression/decompression standard supported by Video for Windows is Cinepak. This standard uses an AVI file format to produce 40:1 compression ratios, and 30-fps capture, at 320×200 resolution.

Windows 95 naturally supports several different compression techniques. These include Cinepak, two versions of Indeo, an RLE format, and the Video 1 format.

Video Capture Software

The video capture software normally provides the user with an interface to configure and conduct a video capture operation. This interface, like most other capture software, enables the user to view audio waveforms and video images, perform file creation and control functions, capture video (single-frame or full-motion), and edit video clips or still frames for content and effects.

The capture interface screen shown offers a control panel for file-handling and editing functions and a monitor window to allow incoming video to be previewed. An audio wave display window, above the control panel, displays the audio waveforms being recorded or played back. This feature is very useful for editing audio files. It is possible to remove sections of the audio and relocate them. The shuttle dial provides quick movement through the file. A set of tape counters at the left side of the panel provide start, length, position, and end values for the video in digital display format.

Normal file-handling features include Open and Save functions. In this example, they are present in a pair of buttons toward the right side of the screen. Drive and filename listings are available in the windows at the extreme right side of the control panel.

Some File Editing functions typically found in video capture software include the following:

- Zoom
- Undo
- Cut
- Paste
- Crop
- Clear

These functions can be used to edit both audio and video files. Typical video effects include preview and freeze. The freeze function captures a still video frame; the preview function allows the video source to be examined, without capturing frames.

A full set of audio and VCR control buttons are located on the control panel, to manipulate the record and playback functions for audio or video. The type of file being examined is determined by the audio/video button that changes from audio to video to AVI.

The compression controls enable the user to select the type of compression format to be used. Other compression control settings are used to set a balance between the size and quality of the AVI file produced. The user can select the target (RAM or HDD) for the captured video stream, determine the number of key frames per second, and set the length of the capture buffer that holds the incoming video information.

Typical capture rates for full-motion video cards are stated at 30 frames per second. However, many systems are not able to reach the full potential of this rate. A number of frames may be dropped from the sequence of frames shot. To record at frame rates faster than 15 fps, the system must have a full arsenal of multimedia-capable components, because successful video capture is normally a function of the system's hardware capabilities. If the system's RAM is being used to buffer the incoming video information, the size of the extended memory section needs to be very large.

Many MPC system include very large, very fast hard drives for data buffering. The hard drive space allows much more video to be captured, provided the HDD is fast enough to keep up with the incoming data stream. Video buffering is one application where even the slightly superior performance of SCSI-2 systems become more desirable than less-expensive EIDE systems.

Installing Video Capture Cards

Refer to the capture card's documentation, and prepare its hardware configuration jumpers (or switches) for operation. The card's factory default settings usually work well, but you should examine the system's installed devices for address and IRQ conflicts.

Install the video capture card in one of the unit's adapter slots. As with any other adapter card, this will require that the outer cover of the system unit be removed. Inside the unit, remove the expansion slot cover of a compatible expansion slot. Many capture cards are full-length cards, so make sure the slot can handle the physical dimension of the card. Make sure the expansion slot type is compatible with the capture card's edge connector.

Connect the capture card to the VGA card as directed by the manufacturer's Installation Guide. Figure 12.30 depicts a BERG connection between the capture card and the VGA feature connector on the VGA card. Some VGA cards use an edge connector built in to the top of the card for the feature connection function.

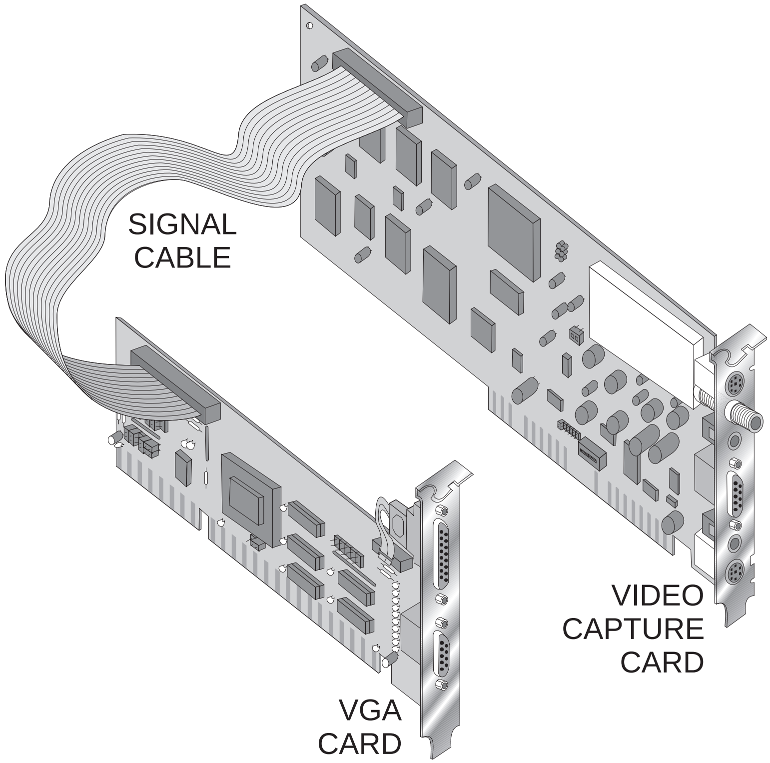


Figure 12.30

Capture card-to-VGA connection.

Install any antennas that need to be connected to the card for the intended application. These could include a TV antenna, a coaxial cable from the television, or an FM radio antenna.

Connect the video-in cables to the audio and video source(s) being used for input. In this case, there is a stereo audio-in (left/right) provision and two possible sources for video-in. The audio and video connections are typically made with standard RCA cables and connectors.

Connect the VGA monitor's signal cable to the capture card's VGA-out connector, as illustrated in Figure 12.31. The video signal passes through the capture card and is looped to the VGA card. This allows the screen image to be present on the monitor and the video screen simultaneously. A VGA loopback cable is connected between the capture card's loopback input and the VGA card's RGB-out connector.

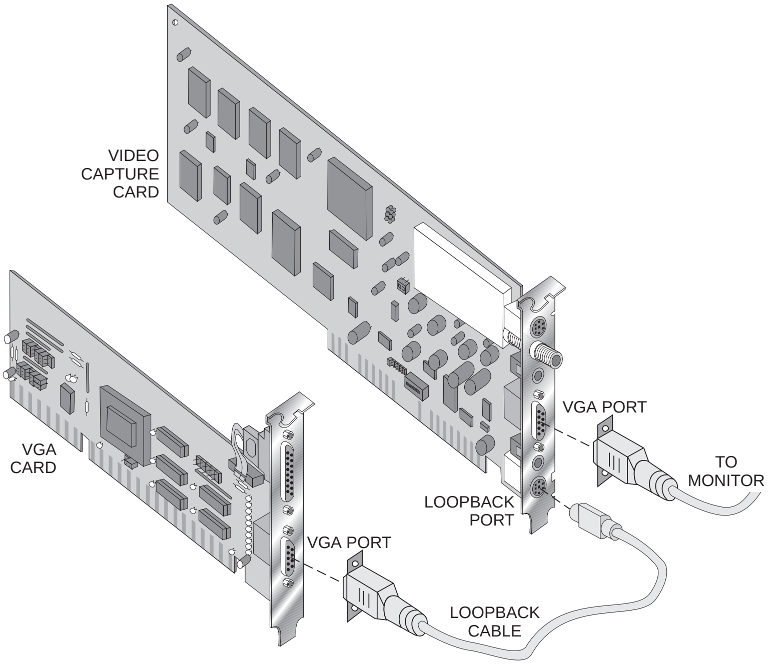


Figure 12.31

Connecting the monitor's signal cable.

If external speakers and microphones are being used, connect the microphone to the MIC input, and the speaker to the speaker port. This port can also be used as an audio source to the line-in connection of a sound card. This connection is described in Figure 12.32. These types of inputs typically use RCA stereo jacks and plugs.

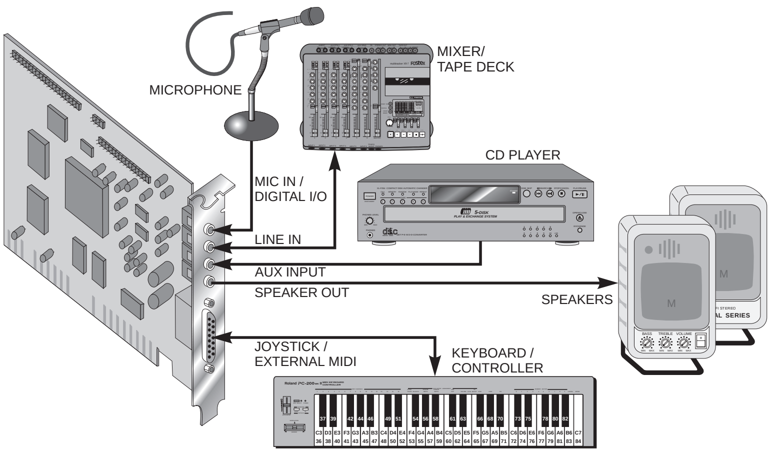


Figure 12.32

Connecting the sound card.

Troubleshooting Video Capture Cards

Troubleshooting problems with a video capture card can involve many parts of the system. Because the capture card can get its input from so many sources, all TV-signal input devices must be considered a possible cause of capture problems. Because the capture card is directly involved with the video card, it must always be considered as a source of video capture problems. Because of the complexity of the possible TV/video/sound components, cabling is always a potential source of problems.

Video Capture Card Configuration Checks

A typical capture card's video address range is between 700_h and $70F_h$. Alternate video address settings usually include $400–40F_h$, $500–50F_h$, and $600–60F_h$. These settings may be made through BERG jumpers or through a multiple-switch set mounted in a Dual Inline Package (DIP), as illustrated in Figure 12.33. Address settings for a particular brand of capture card would be found in its Installation Guide. If the hardware jumper setting is changed on the card, make certain to reset the video capture software configuration to match it. This should be done through the Setup utility of the card's user software.

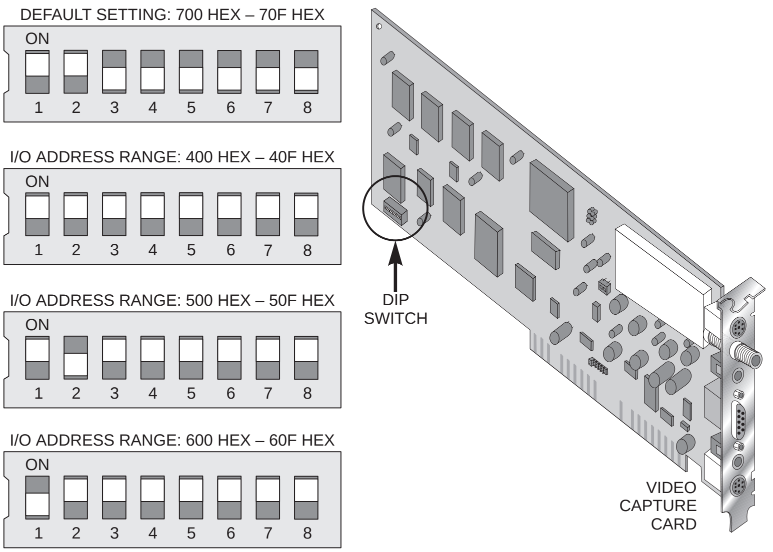


Figure 12.33

Configuring addresses with DIP switches.

As with most other adapter cards, the capture card will need to have an interrupt request channel set up for it. IRQ10 is a typical default setting. However, IRQ6, IRQ11, and IRQ12 are possible alternatives. As with the base address value, the IRQ value will need to be updated in the software if the card's hardware setting is changed.

The capture card's general purpose I/O address must also be configured for correct operation. Capture cards typically use I/O addresses $30C_h$–$30F_h$ or $20C_h$–$20F_h$ for I/O purposes. This is the address space where the capture card's onboard controller communicates with the system. It is typically used when the system initializes the card for use, and when it changes the card's operating parameters. Likewise, the system can obtain status information from the card through these addresses.

Capture cards can contain a number of other, more proprietary, hardware settings. The manufacturer's Installation Guide must be consulted to find out what they are and how they should be set up for a particular application. If the hardware settings are changed from the factory defaults, and power is lost to the system, the software settings may need to be reset to match the true hardware settings.

Video Capture Card Software Checks

No diagnostic utilities are available to test the function of video capture cards. However, a diagnostic package can be run to check for addressing, IRQ, and DMA conflicts. As always, reconfigure the system's components to remove any conflicts detected.

If the capture card's driver software is not installed, or is not installed in the correct directory, the card will not be able to capture video. Check the Installation Manual to verify that the drivers are installed correctly. If problems persist, reinstall the video capture software by running the manufacturer's Installation disk. Contact the capture card manufacturer for updated drivers or driver corrections (referred to as patches).

Check the AVI functions in Windows. Try to play AVI files that have previously worked through the Video for Windows utility. If the display disappears from the screen as soon as the Windows Program Manager appears, check the Startup window for capture card-related EXE files. These files can be REMed for testing purposes.

Check the Drivers window in the Control Panel to see that the card's AVI drivers are properly installed. Consult the manufacturer's User's Manual for the names of the card's AVI drivers.

If the Media Player will not run animations, check in the Device section to make certain that it is set to play animation files. Check to see that the animation driver is installed in the Control Panel's Driver-settings area. See whether the Microsoft MCI Driver for MMP is properly set up.

Under Windows 95, the video capture information is located under the System section in the Control Panel. Check the Device Manager for the capture card's information. Also, check under the Advanced tab of the Control Panel's Multimedia section for driver information concerning the sound card. Check the installed driver and its properties to see that it is set up correctly for the installed capture card.

Video Capture Card Hardware Checks

If all the hardware and software configuration settings are correct, but capture problems continue, it will be necessary to troubleshoot the video capture-related hardware. In most systems, this will involve a TV signal source (VCR, camcorder, and so forth), the cabling, the capture card, and the video card. Other parts of the computer system may become involved, but the components presented are those hardware components typically associated with video capture operations.

Most capture card software provides a preview window that enables the user to view the video coming from the video source. If the source is visible in this window, the video source and the video-in cabling can be eliminated as a source of problems. Just being able to see the video in the window, however, does not mean that the card will capture video.

If the video is present in the window and the video source and cabling check out, the hardware and software configurations should be checked closely. Check the capture software's setup for video buffer settings. Usual video buffer settings are $D0000_h$, $D800_h$, $E000_h$ or $E800_h$. Change the setting to one of the other possible values. Add a `DEVICE=` statement to the CONFIG.SYS file that corresponds to the new setting for the video buffer (that is, `DEVICE=path\EMM386.EXE X=D000h-D700h`). Finally, reinstall and reconfigure the capture software if problems continue.

If the signal from the video source is not present in the preview window, make certain that the video source is turned on. Check the video-in cable to make certain that it is properly connected to the video-out jack of the video source and into the correct video-source input on the capture card. Check the capture card's I/O address setup closely, as well as its setting in the capture software. Check in the video capture software to make sure that the correct video source setting is selected. While in the software settings, check the video type selected and make sure that it is set for the NTSC standard.

VGA-to-TV Converters

Objective

Several companies have developed adapter cards and external adapters to convert VGA signals into National Television Standards Committee (NTSC) and European PAL-compatible video signals. This allows the computer to use a television as an output device. This type of converter is particularly useful for presentations conducted in medium-sized rooms. A large-screen or rear-projection television can produce readable output in such a room without dimming the lights. A very-large-screen VGA monitor would be very expensive by comparison. Using an overhead projector/LCD panel combination can be difficult to read without lowering the room lights. Even with very powerful projectors, the output can be washed-out by normal room lighting.

Figure 12.34 depicts a typical VGA-TV converter card. These converters also come packaged as external units that attach to the output connector of the VGA card. The output signal of the VGA card is routed onto the converter, and re-emerges to be connected to the VGA monitor. The TV connection is usually made through RCA cables that connect to the TV's video-in jacks. Some converters work in parallel with the VGA card, through the IBM PS/2-compatible auxiliary video extension.

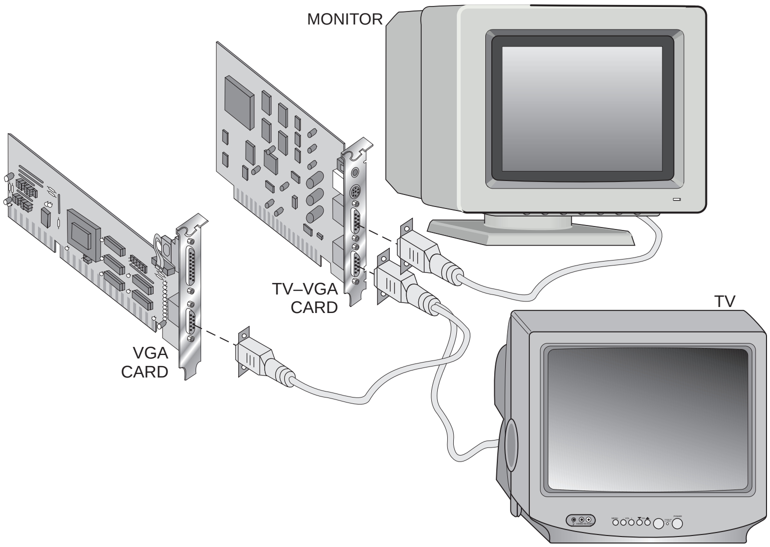


Figure 12.34

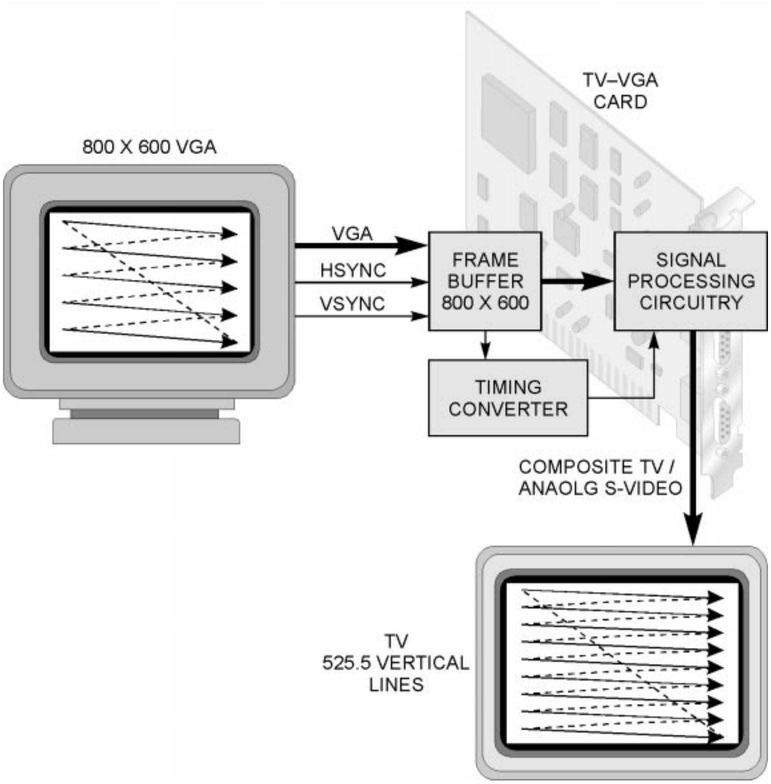
Connecting a VGA-to-TV card into the system.

The internal operation of the converter is described in Figure 12.35. Its operation is straightforward in that its signal-processing circuitry converts the raster-scan video and sync signals of a VGA or SVGA card into an NTSC-compatible raster-scan signal. For example, the converter might be required to convert an 800 dot-by-600 raster-line signal into a 525.5 raster-line—compatible TV signal.

In the converter, the VGA signal is sampled into an onboard memory, called the `frame buffer`. An onboard clock signal outputs the contents of the buffer at TV-compatible rates. The output RGB signals are fed into a composite signal encoder that performs the final signal conversion into composite-TV or analog S-Video format.

Figure 12.35

Internal operations of a converter.



Troubleshooting VGA-to-TV Converters

The VGA-TV card becomes part of the computer's video output system. Therefore, troubleshooting problems with a VGA-TV card typically involve the other parts of the display system. Figure 12.36 illustrates the parts of the video output system, when a VGA-TV card is involved.

In addition to the hardware components of the system, the VGA-TV card's driver software must also be considered as a potential source of problems.

VGA-to-TV Converters Configuration Checks

Refer to the converter's Installation or User's Guides for configuration information. Check for proper jumper settings if the converter uses hardware jumpers. Check the setup of the converter's driver software. Most VGA-TV converters are software configurable.

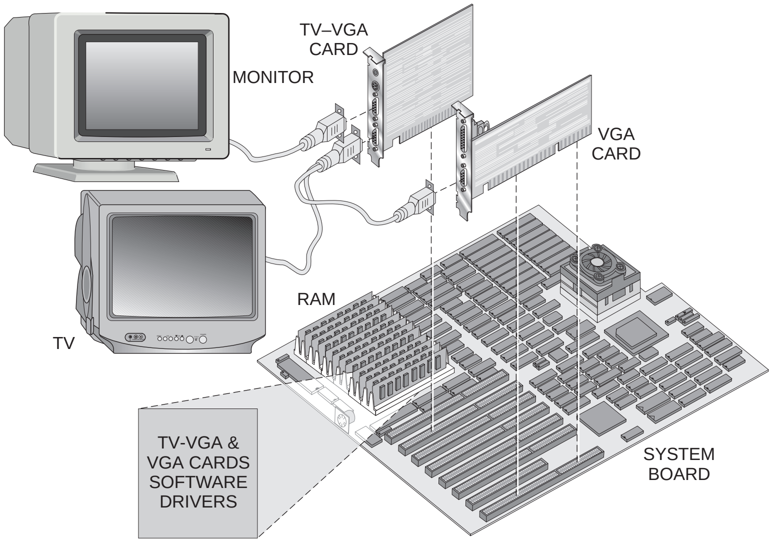


Figure 12.36

The components of the video output system.

VGA-to-TV Converters Software Checks

Make certain that the VGA-TV driver is loaded. The driver should be a TSR type of driver. Run the TSR function of a diagnostic, such as MSD to see whether the system sees the VGA-TV driver. The TSR driver may conflict with other system drivers. This will result in other applications that do not operate properly after the converter has been installed. Refer to the User's Guide for alternative installation guidelines for the driver.

Check in the AUTOEXEC.BAT and CONFIG.SYS files for the conflicting TSRs. Create a clean boot disk with no AUTOEXEC.BAT or CONFIG.SYS files, and then load the converter's driver software from the DOS prompt. If there is a TSR conflict problem, it should disappear when the driver is run in this manner.

If you are operating the converter in Windows, check the Windows video setup for the proper display mode. Some converters will not operate in SVGA mode. If the Windows video is set for SVGA mode, select a 640×480 driver, and retry the system. In some cases, the SVGA mode selection will cause the television display to be jumpy, while the VGA output is steady.

VGA-to-TV Converters Hardware Checks

The equipment associated with a VGA-TV converter includes the following:

- The converter
- The VGA card
- The TV
- The cabling
- The computer system

These components are depicted in Figure 12.36. If the video signal is not appearing on the TV screen, check the cabling and connectors to make sure they are installed properly, as directed by the manufacturer's Installation Guide.

Check the TV's channel setting and on/off switch to make sure that the TV is on and set to the correct channel. Also, check the TV's video input setting to make sure that it is set to match the actual hardware connection between the TV and the VGA converter.

If there is a VCR connected between the TV and the converter, check the TV/VCR channel settings. If the converter is an external unit, make sure that its external power supply is plugged in, and that power is available.

If the computer is a notebook with an auxiliary VGA output (the LCD panel is the primary VGA output), check to make sure that the external VGA switch is set properly. The notebook may also have a software `hot key` that switches the output from the LCD panel to the external VGA output. Consult the notebook's User's Manual for this possibility.

General Multimedia Problems

Typical symptoms associated with multimedia failures include the following:

- Sound not working.
- The system will not capture video.
- Cannot access the CD-ROM.
- The system will not play video.

Various multimedia support systems include a sound card, a CD-ROM drive, external speakers, and a video capture card. Most or all these devices will be included in any particular multimedia system. These types of equipment typically push the performance of the system and, therefore, require the most services from technicians. This is very true during setup and configuration.

One of the major points to be aware of when building or upgrading a multimedia PC is the interrupt usage of the system. It is important that all the devices have access to unique, acceptable interrupt request lines. To ensure this, the technician should map out the system's IRQ capabilities with the number and level of interrupts needed by the different devices being installed. In many instances, it will be necessary to map the DMA capabilities of the system to the number of available DMA channels. The wide variety of I/O systems that come together to create a true multimedia machine can quickly use up all the available I/O slots on most system boards. This is particularly prevalent in Pentium systems, where the expansions slots are often a mix of ISA and PCI buses. This sometimes leads to problems getting I/O cards with the right mix of bus connectors. IRQ and DMA availability and utilization are covered in detail in Chapter 6.

Summary

The focus of this chapter has been on multimedia components. The initial section of the chapter focused on multimedia applications. The Windows multimedia support tools were stressed along with typical multimedia authoring systems.

A large portion of the chapter was dedicated to the hardware side of creating and playing multimedia titles. The operation of CD-ROM drives, sound cards, video capture cards, and VGA-to-TV converter cards was covered in detail. A troubleshooting section was presented along with each topic. Each section contains typical symptoms and troubleshooting procedures.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand each point and can perform each task listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercises

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedure 29—CD-ROM Installation and Setup.

Review Questions

1. How is data stored on a CD?
2. How do CD players differ from CD-ROM drives?
3. List six hardware components typically found in a multimedia system.
4. What is the purpose of an authoring system?
5. Why are fast peripherals desirable with multimedia systems?
6. What type of file is a WAV file?
7. Describe interactivity as it applies to multimedia applications.
8. Describe the two types of video capture cards.
9. Name the two types of video compression standards, and describe the type of application each can be found in.
10. What hardware component is associated with the MSCDEX file?
11. List at least four types of software packages generally associated with the production of a multimedia title.
12. What type of conversion is performed by a VGA-TV card?
13. List the hardware items that should be checked if sound card problems are encountered.
14. In what type of multimedia application are you most likely to find MIDI equipment and software?

15. Describe three hardware configuration settings normally associated with setting up a video capture card.
16. How is a multimedia authoring system different than a word processing or desktop publishing program?
17. How is MIDI data exchanged between system components?
18. What type of standard is YUV?
19. Describe the process that takes place in creating digitized audio.
20. What is the purpose of including a video capture card in a multimedia PC?
21. How is digitized video different than digitized audio?
22. Why is cabling such a cause of problems in video capture cards?
23. What is the major limiting factor in using full-motion video in a multimedia title?
24. Which Windows utility is the primary channel used with the MCI interface?
25. Describe the three compression techniques involved in MPEG compression.

Review Answers

1. In the form of microscopic blisters written on a light-sensitive material by a laser. For more information, see the section titled “CD-ROM.”
2. Information from audio CDs is passed back and forth through A-D and D-A converters; data used with CD-ROM disks always remains in a digital format. For more information, see the section titled “CD-ROM.”

3. Very fast microprocessors, a lot of RAM, large, fast hard-disk drives, fast CD-ROM drives, a sound card, and a high-resolution video card. For more information, see the section titled “Multimedia Hardware.”

4. These software packages are used to pull together various files types from different media and organize them into a cohesive presentation. For more information, see the section titled “Multimedia Applications.”

5. The large sizes associated with digitized audio and video files require fast I/O devices that can handle large files efficiently. For more information, see the section titled “Multimedia Hardware.”

6. A WAV file is a standard audio file. For more information, see the section titled “Multimedia Applications.”

7. The ability of users to participate in the program while it is running. For more information, see the section titled “Multimedia Applications.”

8. Single, or still-frame capture and full-motion capture. Single-frame capture cards “grab” a single frame of information from a video source. Full-motion video capture involves capturing consecutive frames of information from a video source such as a camcorder or VCR. For more information, see the sections titled “Multimedia Hardware,” “JPEG,” and “MPEG.”

9. The JPEG standard supports single-frame capture; the MPEG standard supports full-motion video capture. For more information, see the sections titled “”JPEG,” and “MPEG.”

10. The CD-ROM drive. For more information, see the section titled “Configuring a CD-ROM Drive.”

11. The authoring package, a graphic design package, video capture software, audio digitizing software, and word processors. For more information, see the section titled “Multimedia Applications.”

12. An RGB VGA video signal into a composite TV (raster-scan) video signal. For more information, see the section titled “VGA-to-TV Converters.”
13. The speakers, the sound card, and the amount of free space on the HDD. For more information, see the section titled “Sound Card Hardware Checks.”
14. Electronic music production. For more information, see the sections titled “Windows Multimedia Support” and “Audio Tools.”
15. The video address, the IRQ setting, and the I/O address range. For more information, see the section titled “Video Capture Card Configuration Checks.”
16. Items produced in an authoring system are dynamic. Events in the final product occur in relationship to time and input. Output from a word processor or desktop publishing package are static. The appearance is present immediately. The appearance of a static piece is composed; an authored piece is a staged production. For more information, see the section titled “Multimedia Authoring Tools.”
17. Serially. For more information, see the section titled “MIDI.”
18. A television industry color signal standard. For more information, see the section titled “Capture Card Signals and Connections.”
19. The microphone converts the audio information into an analog electrical signal. The sound card samples this signal periodically through an A-to-D converter. The digital samples produced represent the signal at particular instances of time. For more information, see the section titled “Sound Cards.”
20. To convert video information from different sources into digital information that can be processed by the computer for inclusion in multimedia titles. For more information, see the section titled “Video Capture Cards.”

21. Digitized video cannot just be sampled and compressed. Instead, video information must be monitored from byte to byte and frame to frame for redundant information. For more information, see the sections titled “Video Capture Cards” and “Capture Card Signals and Connections.”

22. Because there are a number of different connections involved in installing a typical video capture card. For more information, see the section titled “Troubleshooting Video Capture Cards.”

23. Data storage space both in RAM and on the HDD. For more information, see the section titled “Multimedia Hardware.”

24. The Windows Media Player. For more information, see the section titled “Audio Tools.”

25. Interframe compression (removing redundant information from consecutive frames), intraframe compression (removing redundant information within a single frame), and audio compression. For more information, see the section titled “MPEG.”

Chapter 13

Preventive Maintenance and Safety

After completing this chapter and its related lab procedures, you should be able to meet the following objectives:

Objectives

- List the steps for proper IC handling.
- Describe ESD hazards and methods of preventing ESD.
- Describe the requirements of a proper work area.
- State typical precautions that should be observed when working on computer equipment.
- Define the term *ground*.
- Describe the two types of uninterruptible power supplies (UPS) and state their qualities.
- Differentiate between various UPS specifications and state how they apply to a given situation.
- Differentiate between total, selective, and differential backup methods.
- List precautionary steps that should be taken when handling floppy disks.
- Detail routine preventive maintenance procedures as they apply to hard and floppy disks.
- List steps to clean a dot-matrix, ink jet, or laser printer.

continues

- Establish and maintain preventive maintenance schedules for users.
- Demonstrate proper cleaning procedures for various system components.
- Perform basic disk-management functions on a hard drive, including using SCANDISK, CHKDSK, and Defrag utilities.
- Use backup software to create backups of important data.
- Describe different types of data backup that can be performed with typical backup utilities.
- Use software utilities to identify and remove viruses from computer systems.
- State potential hazards that are present when working with laser printers, monitors, and so forth.
- Perform generic preventive-maintenance routines as required (that is, remove excess toner, replace printer ribbons, defragment hard drives, create backup copies).

Introduction

It has long been known that one of the best ways to fix problems in complex systems is to prevent them before they happen. This is the concept behind preventive-maintenance procedures. Breakdowns never occur at convenient times. By planning for a few minutes of non-productive activities, hours of repair and recovery work can be avoided. In this chapter, common preventive-maintenance activities and scheduling are covered.

Safety-related issues are presented in this chapter. Although the personal computer is a relatively safe piece of equipment to own, or repair, it is still an electronic device that carries potentially hazardous conditions with it. This short section should make you aware of these potential hazard areas.

Cleaning

Objective

Cleaning is a major part of keeping a computer system healthy. Therefore the technician's toolkit should also contain a collection of cleaning supplies. Along with hand tools, it will need a lint-free, soft cloth (chamois) for cleaning the outer surfaces of the system. Outer-surface cleaning can be accomplished with a simple soap and water solution, followed by a clear-water rinse. Care should be taken to make sure that none of the liquid splashes or drips into the inner parts of the system. A damp cloth is easily the best general purpose cleaning tool for computer equipment.

The cleaning should be followed by the application of an antistatic spray or solution to prevent the build up of static charges on the components of the system. A solution composed of 10 parts water and 1 part common household fabric softener makes an effective and economical antistatic solution. To remove dust from the inside of cabinets, a small paint brush is handy.

Another common problem is the build up of oxidation, or corrosion, at electrical contact points. These buildups occur on electrical connectors and contacts, and can reduce the flow of electricity through the connection. Some simple steps can be used to keep corrosion from becoming a problem. The easiest step in preventing corrosion is observing the correct handling procedures for printed circuit boards and cables, as shown in Figure 13.1. Never touch the electrical contact points with your skin; the moisture on your body can start corrosive action.

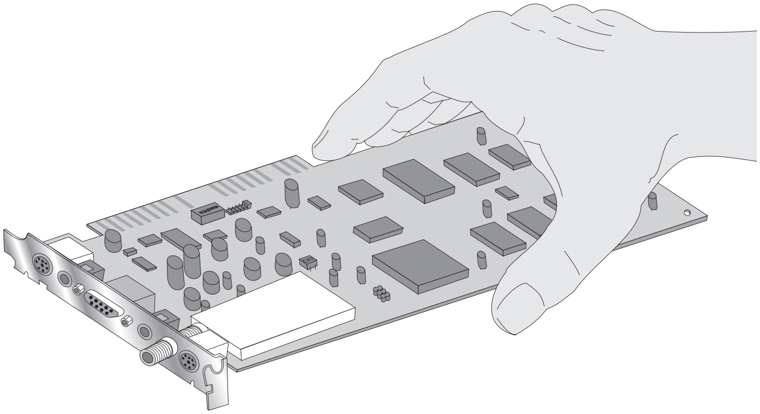


Figure 13.1

How to handle a PC board.

Even with proper handling, some corrosion may occur over time. This oxidation can be removed in a number of ways. The oxide buildup can be sanded off with emery cloth, rubbed off with a common pencil eraser or special solvent-wipe, or dissolved with an electrical-contact cleaner spray. Socketed devices should be reseated as part of an anti-corrosion cleaning. However, they should be handled according to the MOS Handling guidelines in this chapter to make certain that no static discharge damage occurs.

If you use the emery cloth or rubber eraser to clean your contacts, always rub toward the outer-edge of the board or connector to prevent damage to the contacts. Rubbing the edge may lift the foil from the PC board. Printed-circuit board connectors are typically very thin. Therefore only rub hard enough to remove the oxide layer. Also, take time to clean up any dust or rubber contamination generated by the cleaning effort.

Cleaning other internal components, such as disk drive Read/Write heads, can be performed using lint-free, foam swabs and isopropyl alcohol or methanol. It is most important that the cleaning solution be one that dries without leaving a residue. The following lists the tools and equipment recommended for a well-prepared computer repair toolbox:

- Assorted flat-blade screwdrivers
- Assorted Phillips screwdrivers
- Assorted small nut drivers
- Assorted small torx bit drivers
- Needle-nose pliers
- Diagonal pliers
- Contact cleaner
- Foam swabs
- Tweezers
- Cleaning supplies
- Magnifying glass
- Clip leads
- IC extractors

Electrostatic Discharge

Objective

Electrostatic discharges (ESD) are the most severe form of *electromagnet interference* (EMI). The human body can build up static charges that range up to 25,000 volts. These build-ups can discharge very rapidly into an electrically grounded body or device. Placing a 25,000-volt surge through any electronic device is potentially damaging to it. Static can easily discharge through digital computer equipment. The electronic devices used to construct digital equipment are particularly susceptible to damage from ESD. As a matter of fact, ESD is the most damaging form of electrical interference associated with digital equipment.

The following are the most common causes of ESD:

- Moving people
- Low humidity (hot and dry conditions)

- Improper grounding
- Unshielded cables
- Poor connections

Elementary-school teachers demonstrate the principles of static to their students by rubbing different materials together. When people move, the clothes they are wearing rub together and can produce large amounts of electrostatic charge on their bodies. Walking across carpeting can create charges in excess of 1,000 volts. ESD is most likely to occur during periods of low humidity. If the relative humidity is below 50%, static charges can accumulate easily. ESD generally does not occur when the humidity is above 50%. Anytime the charge reaches around 10,000 volts, it is likely to discharge to grounded metal parts.

Although ESD won't hurt humans, it will destroy certain electronic devices. The high-voltage pulse can burn out the inputs of many IC devices. This damage may not appear instantly. It can build up over time and cause the device to fail. Electronic logic devices, constructed from *metal oxide semiconductor* (MOS) materials are particularly susceptible to ESD. The following section describes the special handling techniques that should be observed when working with equipment containing MOS devices.

MOS Handling Techniques

Objective

In general, MOS devices are sensitive to voltage spikes and static electricity discharges. This can cause a great deal of problems when you have to replace MOS devices, especially *complementary-symmetry metal oxide semiconductor* (CMOS) devices. The level of static electricity present on your body is high enough to destroy the inputs of a CMOS device if you touch its pins with your fingers.

To minimize the chances of damaging MOS devices during handling, special procedures have been developed to protect them from static shock. ICs are generally shipped and stored in special conductive plastic tubes or trays. You may want to store MOS devices in these tubes, or you may ensure their safety by inserting the

IC's leads into aluminum foil or black conductive foam (not styrofoam). PC boards containing static-sensitive devices are normally shipped in special, antistatic bags. These bags are good for storing ICs and other computer components that may be damaged by ESD. They are also the best method of transporting PC boards with static-sensitive components.

Professional service technicians employ a number of precautionary steps when they are working on systems that may contain MOS devices. These technicians normally use a *grounding strap* like the one depicted in Figure 13.2. These antistatic devices may be placed around the wrists or ankles to ground the technician to the system being worked on. These straps release any static present on the technician's body, and pass it harmlessly to ground potential. Antistatic straps should never be worn while working on higher-voltage components such as monitors and power supply units. Some technicians wrap a copper wire around their wrist or ankle and connect it to the ground side of an outlet. This is not a safe practice because the resistive feature of a true wrist strap is missing. As an alternative, most technician's work areas include *antistatic mats* made out of rubber or other antistatic materials that they stand on while working on the equipment. This is particularly helpful in carpeted work areas, because carpeting can be a major source of ESD build-up. Some antistatic mats have ground connections that should be connected to the safety ground of an ac power outlet.

To avoid damaging static-sensitive devices, the following procedures help to minimize the chances of destructive static discharges:

1. Because computers and peripheral systems may contain a number of static-sensitive devices, before touching any components inside the system, touch an exposed part of the chassis or the power-supply housing with your finger, as illustrated in Figure 13.3. Grounding yourself in this manner ensures that any static charge present on your body is removed. This technique should be used before handling a circuit board or component. Of course, you should be aware that this technique will only work safely if the power cord is attached to a grounded power outlet. The ground plug on a standard power cable is the best tool for overcoming ESD problems.

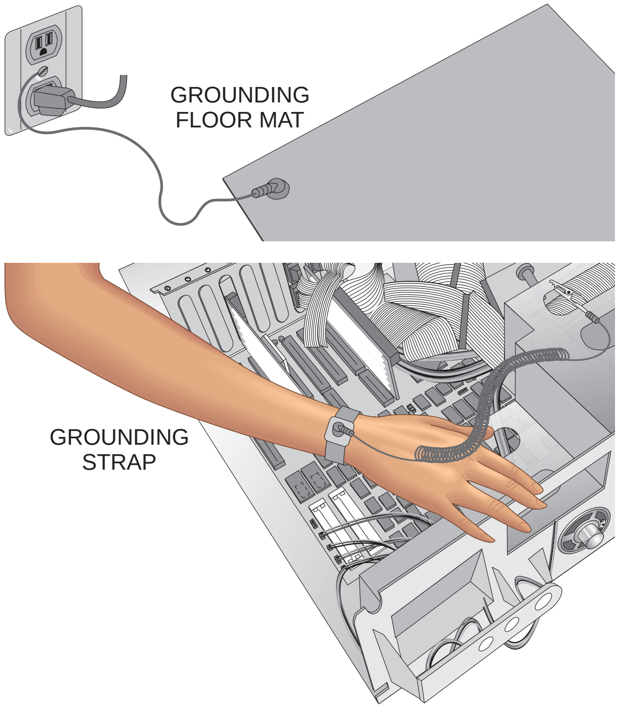


Figure 13.2

Typical antistatic devices.

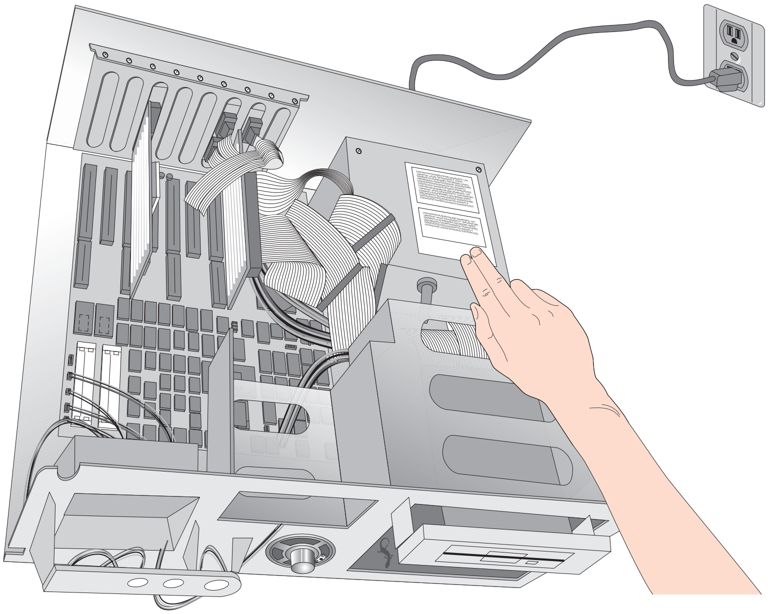


Figure 13.3

Discharging through the power-supply unit.

2. Do not remove ICS from their protective tubes (or foam packages) until you are ready to use them. If you remove a circuit board or component containing static-sensitive devices from the system, place it on a conductive surface, such as a sheet of aluminum foil.

3. If you must replace a defective IC, use a soldering iron with a grounded tip to extract the defective IC, and while soldering the new IC in place. Some of the ICs in computers and peripherals are not soldered to the printed circuit board. Instead, an IC socket is soldered to the board, and the IC is just inserted into the socket. This allows for easy replacement of these ICs. In the event that you have to replace a hard-soldered IC, you may want to install an IC socket, along with the chip. Be aware that normal operating vibrations and temperature cycling can degrade the electrical connections between ICs and sockets over time. Before removing the IC from its protective container, touch the container to the power supply of the unit in which it is to be inserted.

4. Some devices used to remove solder from circuit boards and chips can cause high-static discharges that may damage the good devices on the board. The device in question is referred to as a *solder-sucker*, and is available in antistatic versions for use with MOS devices.

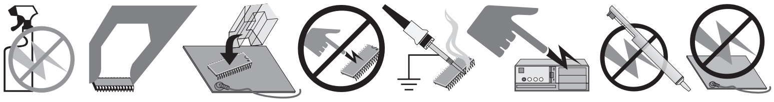
5. Use antistatic sprays or solutions on floors, carpets, desks, and computer equipment. An antistatic spray or solution, applied with a soft cloth, is an effective deterrent to static.

6. Install static-free carpeting in the work area. You can also install an antistatic floor mat as well. Install a conductive tabletop to carry away static from the work area. Use antistatic table mats.

7. Use a room humidifier to keep the humidity level above 50% in the work area. Figure 13.4 summarizes proper IC handling procedures.

Figure 13.4

Antistatic Precautions.



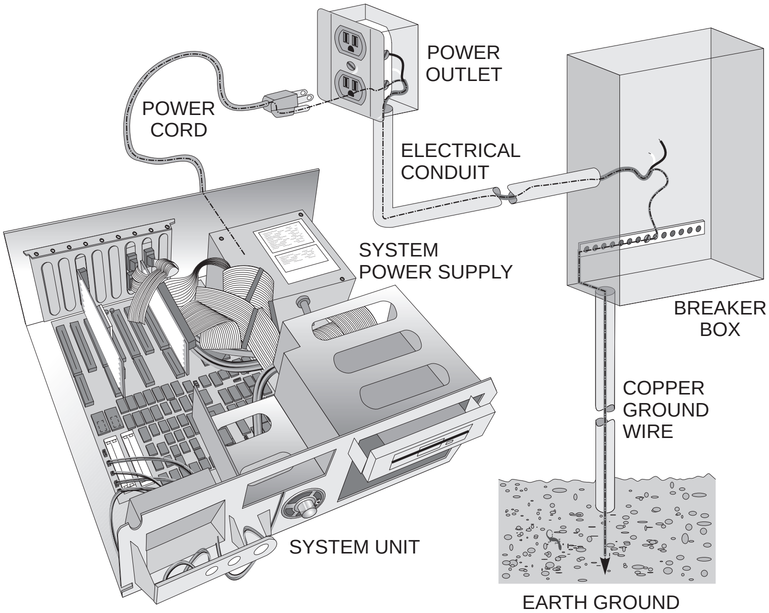
Grounds

Objective

The term *ground* is often a source of confusion for the novice, because it actually encompasses a collection of terms. Generically, ground is just any point from which electrical measurements are referenced. However, the original definition of ground actually referred to the ground, *earth ground* that is. The movement of the electrical current along a conductor requires a path for the current to return to its source. In early telegraph systems and even modern power-transmission systems, the earth provides a return path and, hypothetically, produces an electrical reference point of absolute zero. This type of ground is described in Figure 13.5.

Figure 13.5

Power-transmission system.



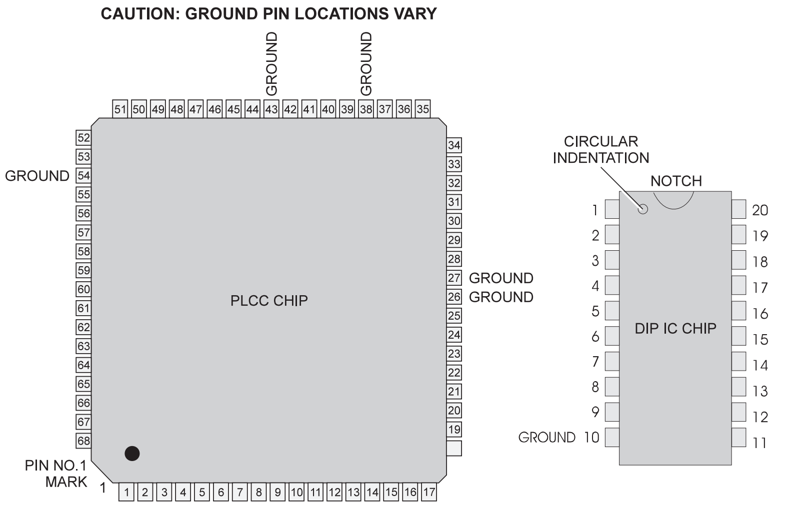
Many electronic circuits use an actual conductor as a return path. This type of ground is referred to as a *signal ground*. Electronic devices may also contain a third form of ground called *chassis*, or *protective ground*. In any event, ground still remains the reference point from which most electrical signals are measured. In the case of troubleshooting computer components, measurements referenced to ground may be made from the system unit's chassis.

The other measurement reference is the signal ground point on the printed-circuit board where the test is being performed. This point isn't too difficult to find in a circuit board full of ICs, because most DIP-style chips use the highest-numbered pin for the positive supply voltage and the last pin on the pin-1 side of the chip as the ground pin. This type of ground is illustrated in Figure 13.6. Some caution should be used with this assumption, however, because not all ICs use this pin for ground. If you examine a number of ICs and connectors on the board, however, you should be able to trace the ground foil and use it as a reference.

As mentioned in Chapter 1, grounding is an important aspect of limiting EMI in computer systems. Left unchecked, EMI can distort images on the video display, interfere with commercial communication equipment such as radios and televisions, and corrupt data on floppy disks. In addition, EMI can cause signal deterioration and loss caused by improper cable routing. If a signal cable is bundled with a power cable, radiation from the power cable may be induced into the signal cable, affecting the signals that pass through it. Unlike ESD, which is destructive, the effects of EMI can be corrected without damage.

Figure 13.6

Grounds on IC chips.



Because the computer system is connected to an actual earth ground, it should always be turned off and disconnected from the wall outlet during electrical storms. This includes the computer and all of its peripherals. The electrical pathway through the computer equipment can be very inviting to lightning on its way to earth ground. The extremely high electrical potential of a lightning strike is more than any computer can withstand.

Power-Line Protection

Digital systems tend to be sensitive to power variations and losses. Even a very short loss of electrical power can shut a digital computer down, resulting in a loss of any current information that has not been saved to a mass storage device. Typical power-supply variations fall into two categories:

- *Transients*—An over-voltage condition. Sags, on the other hand, are under-voltage conditions. Over-voltage conditions can be classified as *spikes* (measured in nanoseconds) or as *surges* (measured in milliseconds).
- *Sags*—Can include *voltage sags* and *brownouts*. A voltage sag typically lasts only a few milliseconds; a brownout can last for a protracted period of time.

Inexpensive *power line filters*, also called *surge suppressers*, are good for cleaning up "dirty" commercial power. These units passively filter the incoming power signal to smooth out variations. There are two factors to consider when choosing a surge suppresser:

- Clamping speed
- Clamping voltage

These units will protect the system from damage, up to a specified point. However, large variations, such as surges created when power is restored after an outage, can still cause considerable data loss and damage. In the case of startup surges, making sure that the system is turned off, or even disconnected from the power source, until after the power is restored is one option. In the case of a complete shutdown or a significant sag, the best protection for saving programs and data is an *uninterruptible power supply* (UPS).

Uninterruptible Power Supplies

Objective

Uninterruptible power supplies are battery-based systems that monitor the incoming power and kick in when unacceptable variations occur in the power source. The term *UPS* is frequently used to describe two different types of power backup systems. The first is a *standby power system*, and the second is a truly *uninterruptible power system*. A typical UPS system is depicted in Figure 13.7.

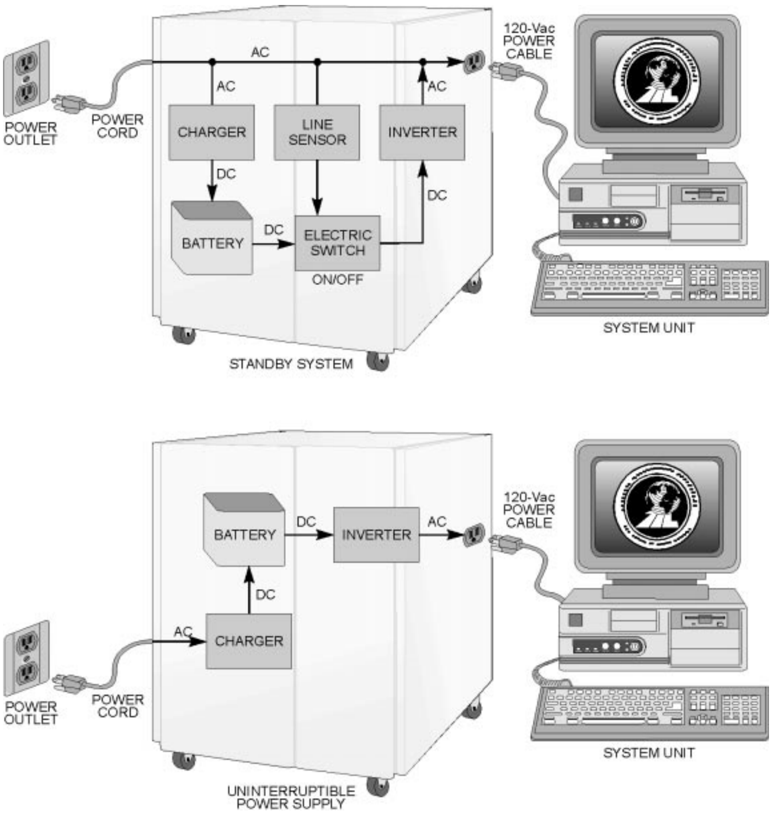


Figure 13.7

UPS systems.

The standby system monitors the power input line and waits for a significant variation to occur. The batteries in this unit are held out of the power loop, and only draw enough current from the AC source to stay recharged. When an interruption occurs, the UPS senses it and switches the output of the batteries into an inverter circuit that converts the DC output of the batteries into an AC current and voltage that resembles the commercial power supply. This power signal is typically applied to the computer within 10 milliseconds.

The uninterruptible systems do not keep the batteries offline. Instead, the batteries and converters are always actively attached to the output of UPS. When an interruption in the supply occurs, no switching of the output is required. The battery/inverter section just continues under its own power.

Standby systems don't generally provide a high level of protection from sags and spikes. They do, however, include additional circuitry to minimize such variations. Conversely, an uninterruptible system is an extremely good power-conditioning system. Because it always sits between the commercial power and the computer, it can supply a constant power supply to the system.

When dealing with either type of UPS system, the most important rating to be aware of is its *volt-ampere* (VA) *rating*. The VA rating indicates the capability of the UPS system to deliver both voltage (V) and current to the computer, simultaneously. This rating is different from the device's *wattage rating*, and the two should not be used interchangeably. The wattage rating is a factor of multiplying the voltage and current use, at any particular time, to arrive at a power-consumption value. The VA rating is used in AC systems because peak voltage and current elements do not occur at the same instant. This condition is referred to as being *out-of-phase* with each other, and makes it slightly more difficult to calculate power requirements. In general, always make sure that the UPS system has a higher wattage capability than the computer requires and, likewise, that the VA rating of the UPS is higher than that required by the computer. High power consumption peripheral devices, such as laser printers, should not be connected directly to the UPS. These devices can overload the UPS and cause data loss.

The other significant specification for UPS systems is the length of time they can supply power. Because the UPS is a battery-powered device, it uses an *ampere-hour rating*. This is the same time-notation system used for automobile batteries and other battery-powered systems. The rating is obtained by multiplying a given current drain from the battery for a given amount of time. (For example, a battery capable of sustaining 1.5 amps of output current for 1 hour would be rated at 1.5 amp-hours.)

Preventive Maintenance

The environment around a computer system, and the manner in which the computer is used, determines greatly how many problems it will have. Occasionally, dedicating a few moments of care to the computer can extend its *mean time between failures* (MTBF) period considerably. This activity, involving maintenance not normally associated with a breakdown, is called *preventive maintenance* (PM). The following sections of this chapter describe PM measures for the various areas of the system.

As with any electronic device, computers are susceptible to failures caused by dust build-up, rough handling, and temperature extremes. Dust build-up can be taken care of by cleaning and inspecting procedures discussed earlier. Likewise, conquering rough handling is an easy matter of adjusting practices. However, identifying and controlling heat build-up problems can require some effort and planning. Microcomputers are designed to run at normal room temperatures. If the ambient temperature rises above about 85 degrees F, head build-up can become a problem. High humidity can also lead to heat-related problems.

To combat heat problems, make sure that the area around the system is uncluttered so that free air flow around the system can be maintained. Make sure the power supply's fan is operational. If it is not, replace the power supply unit. Likewise, be sure that the microprocessor fan is plugged in and operational. It is very easy for a high-speed microprocessor to fry if its fan fails. A good rule of thumb is to install a fan on any microprocessor running above 33 MHz.

If heat build-up still exists, check to make sure that the outer cover is secured firmly to the machine and that all of the expansion slot covers are in place. These items can disrupt the designed air flow characteristics of the case. Finally, add another case fan to draw more air through the system unit.

Monitors

The PM associated with video-display monitors basically consists of periodic cleaning, dusting, and good, common-sense practices around the monitor. The monitor's screen and cabinet should be dusted frequently and cleaned periodically. Dust and smoke particles can build up very quickly around the monitor's screen because of the presence of static charges on its face. When cleaning the screen, some caution should be used to avoid scratching its surface and—in the case of antiglare screens—preserve its glare-reduction features.

Aerosol sprays, solvents, and commercial cleaners should be avoided because they can damage the screen and cabinet. The simple cleaning solution described earlier is also fine for cleaning the monitor. Make sure that the monitor's power cord is disconnected from any power source before washing. The monitor's screen should be dried with a soft cloth after rinsing.

The monitor should not be left on for extended periods with the same image displayed onscreen. Over a period of time, the image will become permanently "burnt" into the screen. If it is necessary to display the same information onscreen for a long period of time, turn the intensity level of the monitor down or install a screen saver program to alter the screen image periodically.

Inside the monitor's housing are very dangerous voltage levels. Therefore you should only remove the monitor's outer cabinet if you are fully qualified to work on CRT-based units. Even if the monitor has been turned off and unplugged for a year, it may still hold enough electrical potential to be deadly.

Hard-Disk Drives

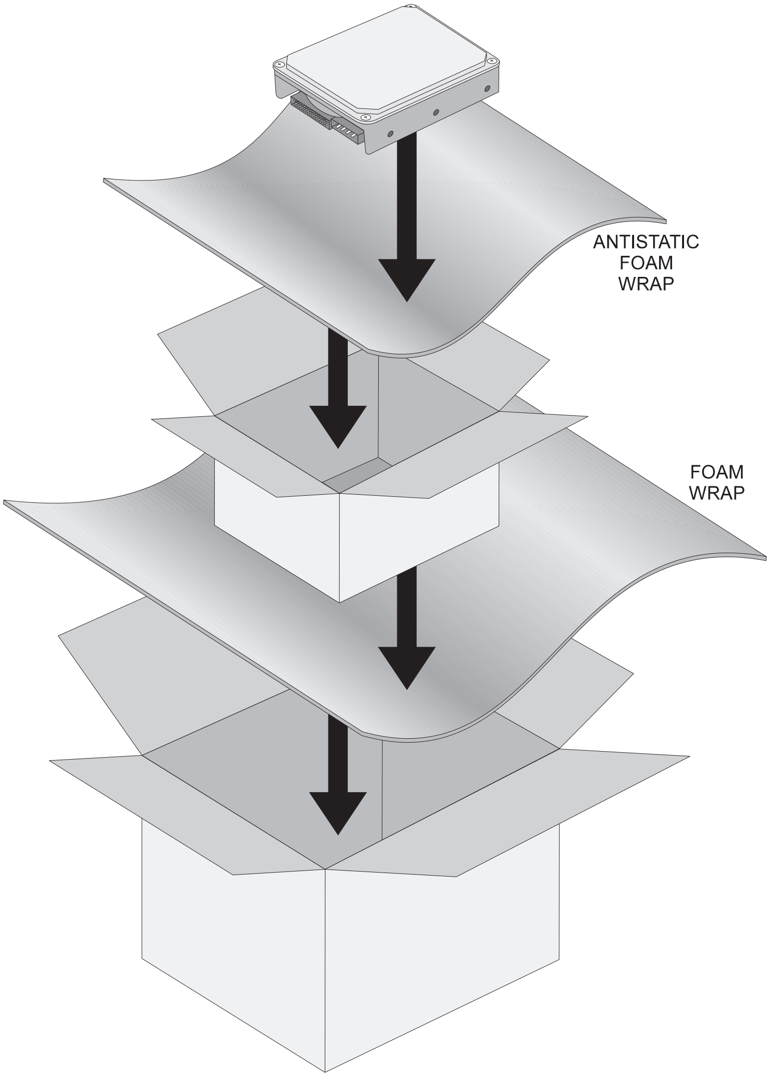
Objective

Hard-disk drives don't require much preventive maintenance, because the R/W heads and disks are enclosed in sealed, dust-tight compartments. However, there are some things that can be done to optimize the performance and life span of hard-disk systems. Rough handling is responsible for more hard-disk drive damage than any other factor. The drive should never be moved while you can still hear its disks spinning. The disk is most vulnerable during startup and shutdown, when the heads are not fully flying. Even a small jolt during these times can cause a great deal of damage to both the platters and the R/W heads. If the drive must be moved, a waiting period of one full minute should be allowed after turning the system off.

If the drive is to be transported or shipped, make sure to pack it properly. The forces exerted on the drive during shipment may be great enough to cause the Read/Write heads to slap against the disks, causing damage to both. Pack the drive unit in an oversized box, with antistatic foam all around the drive. You may also pack the drive in a box-within-a-box configuration, once again using foam as a cushion. This concept is illustrated in Figure 13.8.

Figure 13.8

Proper packing of a hard drive for shipment.



The mechanical mechanisms of the drive should never be positioned by hand. The weighted drive spindle and the track-zero sensor may be accessible with the drive's outer cover removed. Moving these mechanisms by hand can cause the R/W heads to drag across the disk, damaging the heads and disk surfaces. At no time should the internal housing, which protects the platters, be removed in open air. The contaminants in normal air will virtually ruin the drive. If the drive malfunctions, the electronic circuitry and connections may be tested. But when it comes to repairs within the disk chamber, factory service or a professional service facility with a properly clean room is a must!

To recover quickly from hardware failures, operator mistakes, and acts of nature, some form of software *backup* is essential with a hard-disk system. The most common backup for larger systems is high-speed, streaming-tape cartridges. These can automatically back up the entire contents of the disk drive on magnetic tape. In the event of data loss on the disk, a full re-install from the tape is possible in a matter of a few minutes.

Backup may also be kept on disks. However, the volume of data stored on modern hard disks requires a tremendous number of floppy disks to back up. The floppy disks must also be stored. Other high-volume, disk-based devices such as *optical drives* and *removable hard drives* have become attractive methods for backing up the contents of large hard drives. Various backup methods are depicted in Figure 13.9. In any case, failure to maintain backups will eventually result in a great deal of grief when the system goes down because of a hardware or software failure.

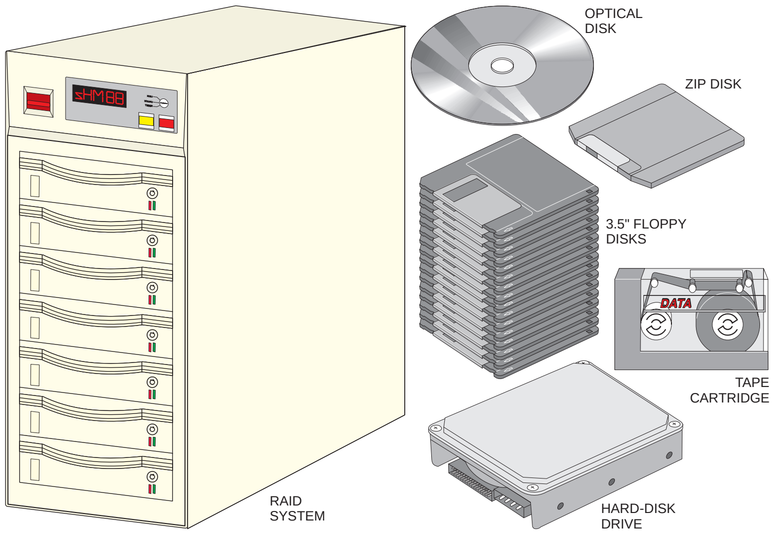


Figure 13.9

Data backup systems.

HDD Utilities

Objective

The operation of hard drives can slow down with general use. Files stored on the drive may be erased, and moved, causing parts of them to be scattered around the drive. This causes the drive to reposition the R/W heads more often during read and write operations, thereby requiring more time to complete the process.

There are five important utilities that can be used to optimize and maintain the operation of the hard disk drive. These are the CHKDSK, SCANDISK, DEFRAG, BACKUP, and Antivirus utilities. With the exception of SCANDISK, which is a Windows utility, all of these utilities have been available since early MS-DOS versions.

In Windows 3.x, these functions were typically located in the Applications window. In Windows 95, they are located in several areas of the system. The icons for SCANDISK and DEFRAG are located in the Program\Accessories\System tools path. The executable file for SCANDISK can be found in C:\Windows. The Backup icon is in the Program\Accessories. the built-in antivirus function is missing from Windows 95. An add-on program from a second party should be used. The MSAV and MWAV programs from DOS and Windows 3.x can be found in the C:\DOS directory if Windows 95 was installed as an upgrade.

CHKDSK

DOS systems offer a number of commands that can be used to maintain and optimize the performance of the hard drive. The DOS `CHKDSK` (Check Disk) command is used to recover *lost allocation units* from the hard drive. These lost units occur when an application terminates unexpectedly. Over a period of time, lost units can pile up and occupy large amounts of disk space. To remove these lost units from the drive, a `/f` modifier is added to the command so that the lost units will be converted into files that can be investigated and removed. In some cases, the contents of the converted file is a usable data file that can be rebuilt for use with an application. The `CHKDSK /f` command is often used before running a drive defragmentation program.

SCANDISK

A similar program, called SCANDISK, is available in DOS 6.x and Windows 95. SCANDISK searches the disk drive for disconnected file clusters and converts them into a form that can be checked and manipulated. This enables the user to determine whether there is any information in the lost clusters that can be restored. SCANDISK also detects and deletes *cross-linked files*. Cross-linked files occur when information from two or more files are mistakenly stored in the same sector of a disk.

The standard SCANDISK operation examines the system's directory and file structure. However, a thorough SCANDISK option can be selected to examine the physical disk surface, as well as its file and directories. If potential defects exist on the surface, SCANDISK can be used to recover data stored in these areas.

DOS and Windows offer a number of utility programs that enable the user to periodically clean up the drive and ensure its top performance. Among these programs are the Defrag, Backup, and Antivirus utilities.

Backup Utilities

Objective

Backup utilities enable the user to quickly create extended copies of files, groups of files, or an entire disk drive. This operation is normally performed to create backup copies of important information in case the drive crashes or the disk becomes corrupt. The DOS Backup and Restore commands can be used to back up and retrieve one or more files to another disk.

Because a backup of related files may be much larger than a single floppy disk, serious backup programs allow information to be backed up to a series of disks. As well, these programs provide file *compression techniques* to reduce the size of the files stored on the disk. Of course, it is impossible to read or use the compressed backup files in this format. To be usable, the files must be *decompressed* (expanded) and restored to the DOS file format.

Backup Types

Objective

Most backup utilities allow backups to be performed in a number of ways. Typically, backups fall into three categories:

- Total
- Selective
- Differential (or modified-only)

In a *full backup*, the entire contents of the designated disk are backed up. This includes directory and subdirectory listings and their contents. In a *selective backup* operation, the operator moves through the tree structure of the disk marking, or tagging, directories and files to be backed up. After all the desired directories/files have been marked, they are backed up in a single operation. Specifying a *differential backup* causes the backup utility to examine each file to determine whether it has changed since the last backup was performed. If the file has been altered, it will be backed up. If not, it is bypassed. This option is a valuable time-saving feature in a periodic backup strategy.

In DOS, the basic backup command can be modified through command switches. An `/s` switch causes all files and subdirectories to be backed up. The `/m` switch modifies the command so that only those files that have changed are backed up. The `/d` and `/t` switches examine the date and time stamps of each file and back up only those files modified after a specified date or time. Other switches can be used to format the backup media and to maintain a backup log on the disk.

Data Backup

From the C:\>> prompt, use the `CHKDSK/f` command to clean up lost file clusters. Instruct the program to convert any *lost chains* into files that can be checked later. Windows 3.x and Windows 95 use a program called MWBACKUP for data conservation on floppy disks.

The operation of the Microsoft Windows 3.x Backup utility is described in the following paragraphs.

Start the backup program. Select the MWBACKUP icon to start the backup program. The MWBACKUP main screen should appear, similar to that shown in Figure 13.10.

Set the backup parameters. Move through the Backup Type drop-down selection box, and select the Full or a partial entry to highlight it, and click on the C: entry in the Backup From list box. Finally, click on the desired drive and disk size entries in the Backup To list box.

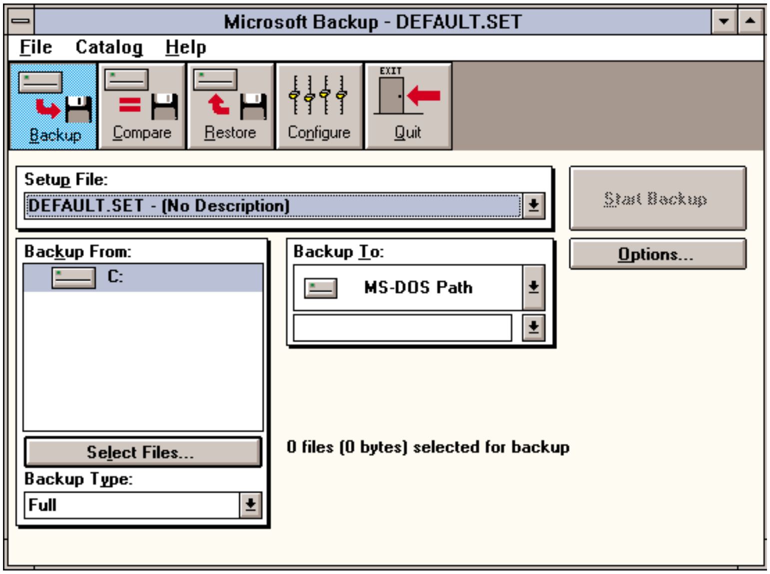


Figure 13.10

The Windows Backup main screen.

Select the backup options by clicking on the Options button. This will provide access to the Backup Options screen depicted in Figure 13.11. Select the desired backup options from the window by clicking on the box next to each option. An *x* will appear beside each item when it is selected.

Select directories and files to back up. This is accomplished by double-clicking on the Select Files button. To choose files to be included in the backup, click on the C:\ folder at the top of the files tree. Click on the Include button, click on the Add button, and click on the Include circle.

Choose special files to be marked by clicking on the Special button, and then clicking on various parameters in the Special Selections dialog box.

Back up the desired portions of the HDD to a series of backup disks. Click on the Backup box to highlight it. Then click on the Select Files button, click on File from the drop-down menu, and click on Select All. Finally, click on the Start Backup box, answer the FDD prompt, and insert a blank disk in the A: drive. The program will conduct the backup operation with no further action from the user, except for exchanging floppy disks when prompted.

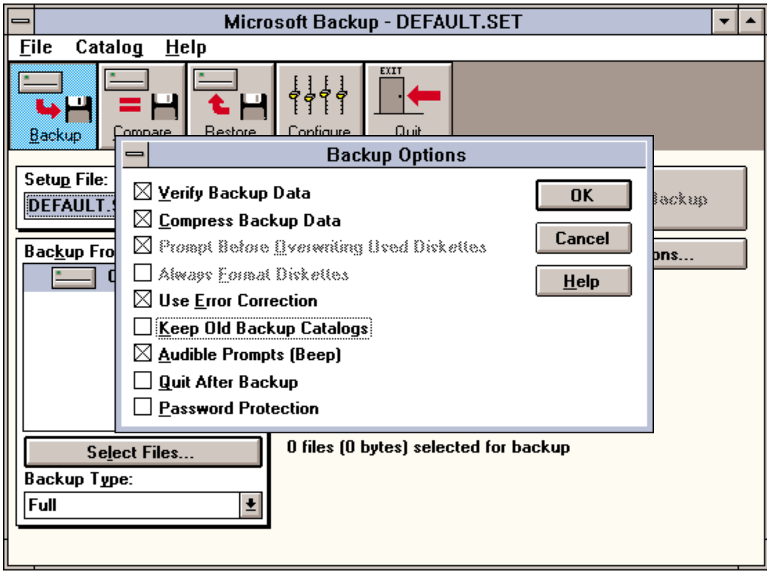


Figure 13.11

The Backup Options screen.

Restoring Data

To restore data in Windows, start the MWBACKUP program. The program's main screen should appear, as previously shown in Figure 13.11.

Set the restore parameters by clicking on the Restore box. Next, verify the floppy disk type in the Restore From window, and verify the C: entry in the Restore Files window. Click on the Restore To window to select it, and then select the location for the restore position.

Select the restore options. Click on the Options button to access the Restore Options screen depicted in Figure 13.12. Select the restore options from the window by clicking on the box next to each option. An *x* will appear beside each item when it is selected.

Restore the specified directories and files to the HDD from the backup disks. Click on the Select Files button. Click on the C:\ file folder to select it. Click on File from the drop-down menu, and click on Select All. Click on the OK button to return to the main screen, and then click on the Start Restore box to begin the restore process.

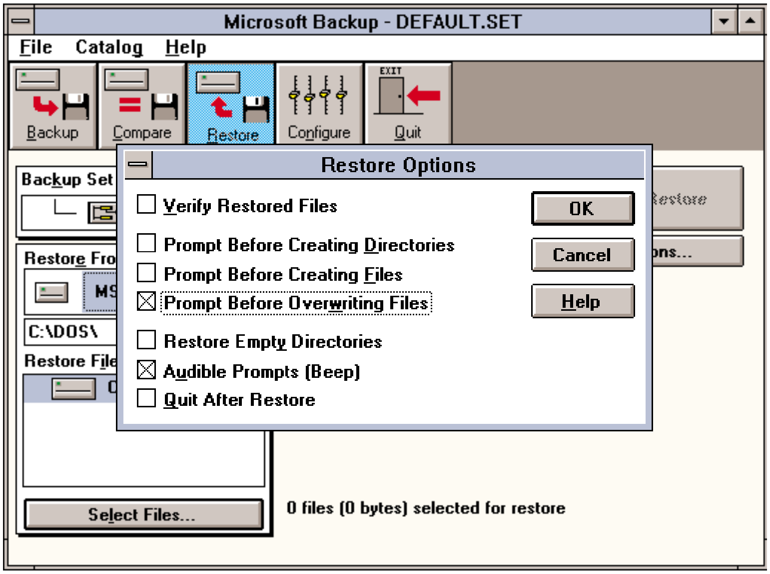


Figure 13.12

The Restore Options screen.

Other Backup Methods

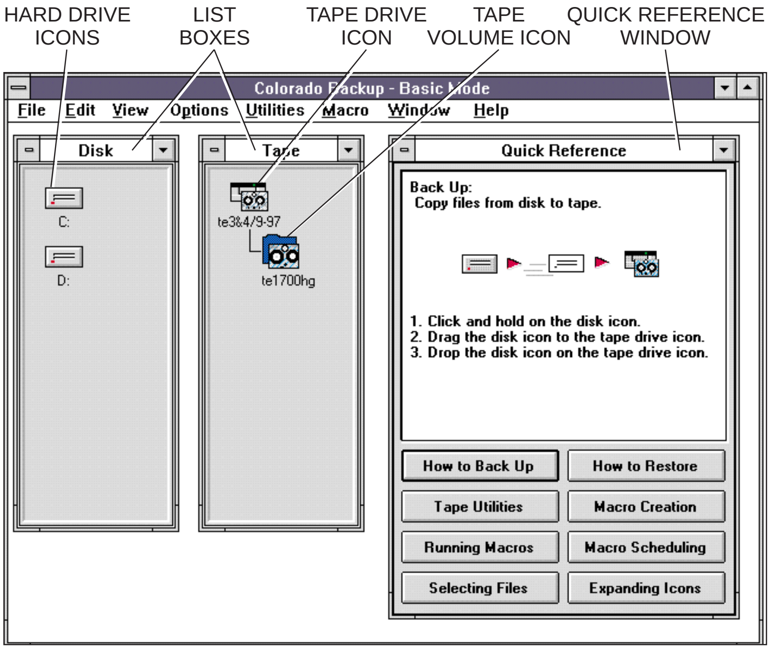
The backup utilities included in DOS and Windows are not the only, or necessarily the best, backup options available. Using DOS or Windows backup utilities can create a huge pile of floppy disks if a large HDD unit is being backed up. Several backup systems using different media have been developed for use with personal computer systems. The most noteworthy are RAID systems, magnetic tape, optical disks, and digital audio tape (DAT) backup systems. Many of these systems have already been discussed in earlier chapters.

In most larger backup operations, it is common to use a tape drive for storage because of its relatively low cost-per-bit of storage.

Figure 13.13 depicts the opening screen of a typical tape backup GUI package. As you can see, this screen incorporates many of the Windows compatibility functions. The toolbar at the top of the screen provides the usual assortment of Windows menu utilities. On the left-hand side of the main screen is a graphical listing of the system's disk drives. The window in the middle of the screen displays the contents of the tape currently in the tape drive. The icon structure allows the contents of drives and tape to be moved and copied using the normal Windows drag-and-drop techniques.

Figure 13.13

A Tape Drive's main screen.



The software's Utility screen, depicted in Figure 13.14, enables users to specify parameters for the backup and restore operations. such as Copy or Move, Total, Selective or Modified. From this screen, users can also choose an immediate or scheduled start of the operation. An additional screen area allows for providing a volume title to the stored information and setting up data compression to reduce the amount of tape required to store the data.

If a selective backup is selected, a file manager screen like the one in Figure 13.15 will appear on the display. This screen is used to *tag* (mark) all the files and directories desired for inclusion in the backup operation.

Most effective tape-storage routines involve rotating multiple tapes to record important data from different days of the week. In a three-tape operation, a relatively small number of tapes can be used to ensure the availability of relatively recent backup data. If only one tape is used, the backup data would be destroyed if the system crashed during the backup. A two-disk rotation creates too much wear on the tapes, causing them to fail prematurely.

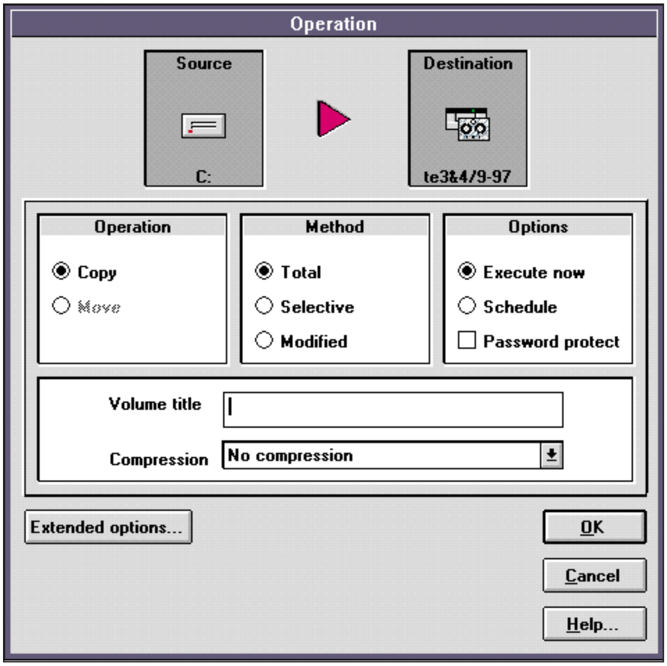


Figure 13.14

The tape drive's Utilities screen.

HDD Defragmentation

In the normal use of the HDD, files become fragmented on the drive, as illustrated in Figure 13.16. This file fragmentation creates conditions that cause the drive to operate slower. Fragmentation occurs when files are stored in noncontinuous locations on the drive. This will happen when files are stored, retrieved, modified, and rewritten based on the differences in the sizes of the before and after files. Because the fragmented files do not allow efficient reading by the drive, it takes longer to complete multisector read operations. The defragmentation program realigns the positioning of related file clusters to speed up the operation of the drive.

Some portions of files may become lost on the drive when a program is unexpectedly interrupted (by a software crash or a power failure, for example). These lost allocation units (*chains*) will also cause the drive to operate slowly. Therefore, it is customary to use the DOS `CHKDSK` command to find these chains and remove them before performing a Defrag operation.

It may also be necessary to remove some data from the drive in order to defragment it. If the system is producing "Out of Disk

Space" error messages, the Defrag utility will not have enough room on the drive to realign clusters. When this happens, some of the contents of the drive will need to be transferred to a backup media (or discarded) in order to free up some disk space for the realignment process to occur.

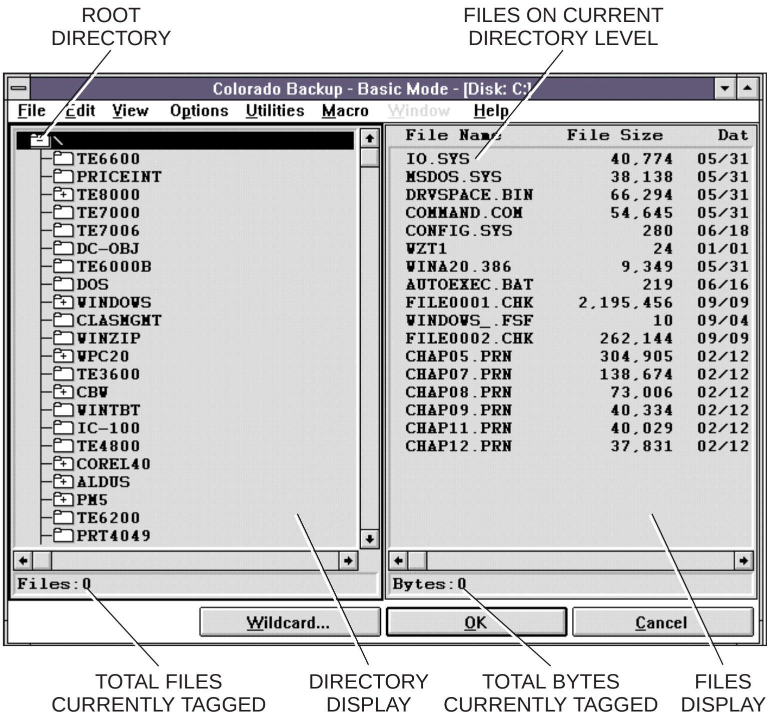


Figure 13.15

Tagging selected files for backup.

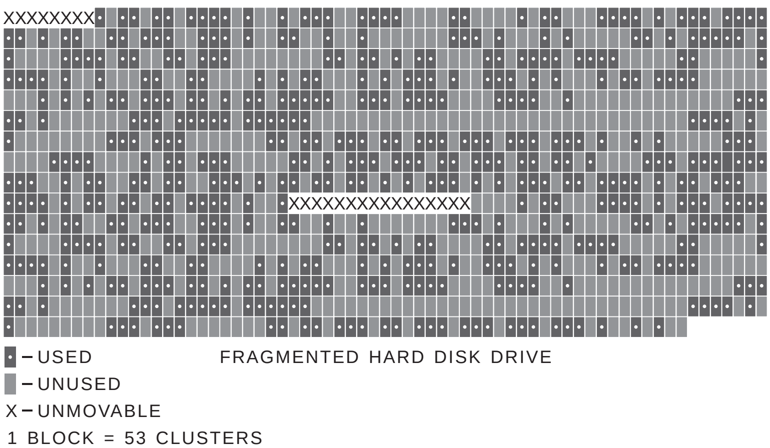


Figure 13.16

Data sectors.

The operation of the MS-DOS Defrag utility proceeds as follows:

1. Start the Defrag program by typing **DEFRAG** at the C:\>> prompt. The DEFRAG main screen should appear, similar to that shown in Figure 13.17.

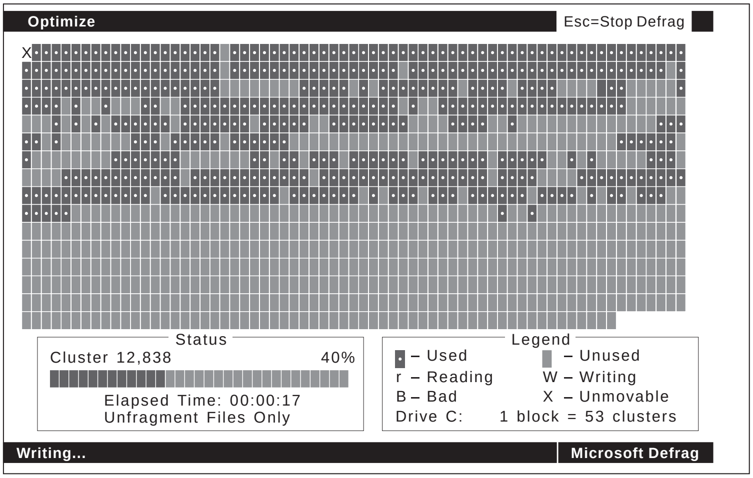


Figure 13.17

The DEFRAG main screen.

2. Select the drive to optimize by highlighting the C: drive and pressing the Enter key.
3. Specify the type of optimization. Highlight Configure, and press Enter. Select the Optimization Method, and select the Unfragment Files Only entry.
4. To specify the type of optimization, press F10, and access the drop-down menu. Use the Down Arrow key to select Optimization Method. Use the spacebar to select Full Optimization. Start the defragmentation procedure by selecting the Begin Optimization entry.

Viruses

Objective

Computer viruses are destructive software programs designed to replicate and spread on their own. Viruses are created to sneak into personal computers. Sometimes these programs take control of a machine to leave a humorous message, and sometimes they destroy data. After they infiltrate one machine, they can spread into other computers through infected disks that friends and co-workers pass around, or through local and wide area network connections.

Researchers at the National Computer Security Association estimate that between 200 and 300 new viruses are being introduced into the computer community every month. Although that number seems alarming, it is the top 10 viruses in the United States that account for about 80% of virus infections.

Most viruses are written for DOS. Therefore, as more and more users switch to Windows, it is expected that the number of viruses being written will decrease. There are basically three types of viruses, based on how they infect a computer system.

One type of virus is known as a *boot sector virus*. This type of virus copies itself on to the boot sector of floppy disks and hard disks. The virus replaces the disk's original boot sector code with its own code. This allows it to be loaded into memory before anything else is loaded. Once in memory, the virus can spread to other disks. Another type of virus is known as a *file infector*. File infectors are viruses that add their virus code to executable files. After the file with the virus is executed, it spreads to other executable files.

A similar type of virus, called a *macro* virus, hides in the macro programs of word processing document files. These files can be designed to load when the document is opened, or when a certain key combination is entered. In addition, these types of viruses can be designed to stay resident in memory after the host program has been exited (similar to a TSR program), or may simply stop working when the infected file is terminated.

The third type of virus is known as a *trojan horse***.** This type of virus appears to be a legitimate program which may be found on any system. Trojan-horse viruses are more likely to do damage by destroying files, and can cause physical damage to disks.

A number of different viruses have been created from these three virus types. They have several different names, but they all perform basically the same. Once the virus file has become active in the computer, it may perform a number of different types of operations. These operations can be as complex and damaging as the author designs them to be.

As an example, a strain of boot sector virus known as *CMOS virus*, infects the hard drive's master boot record and becomes memory resident. When activated, the virus writes over the system's configuration information in the CMOS area. Part of what gets over written is the HDD and FDD information. Therefore, the system will not be able to boot up properly. The initial infection comes from booting from an infected floppy disk. The virus over-writes the CMOS once in every 60 bootups.

A similar boot sector virus, referred to as the *FAT virus*, becomes memory resident in the area of system memory where the IO.SYS and MSDOS.SYS files are located. This allows it to spread to any non-write protected disks inserted into the computer. In addition, the virus moves the system pointers for the disk's executable files to an unused cluster and rewrites the pointers in the FAT to point to the sector where the virus is located. The result is improper disk copies, inability to backup files, large numbers of lost clusters, and all executable files being cross-linked with each other.

In another example, a file infector virus strain, called the *FAT Table virus*, infects .EXE files but does not become memory resident. When the infected file is executed, the virus rewrites another .EXE file that is stored in memory.

Virus Symptoms

Because virus programs tend to operate in the background, it is sometimes difficult to realize that the computer has been "infected." Typical virus symptoms include the following:

- Hard disk control failures.
- Disks that continue to be full even when files have been deleted.
- Cannot read write-protected disks.
- Hard disk stops booting and files are corrupted.
- System will boot to floppy, but will not access the HDD. An `Invalid drive specification` message is usually displayed when attempting to access the C: drive.
- CMOS settings continually revert to default even though the system board battery is good.
- Files change size for no apparent reason.
- System operation slows down noticeably.

- Blank screen when booting (flashing cursor).
- Windows crashes.
- The hard drive is set to DOS compatibility and 32-bit file access suddenly stops working.
- Network data transfers and print jobs slow down dramatically.

A few practices increase the odds of a machine being infected by a virus. These include use of shareware software, software of unknown origin, or bulletin board software. One of the most effective ways to reduce these avenues of infection is to buy shrink-wrapped products from a reputable source.

Another method of dealing with virus protection involves installing a virus scanning program that will check disks and files before using them in the computer. MS-DOS provides a minimal anti-virus scanner called VSAFE that can be installed as a TSR to continuously monitor the system for viruses. DOS also includes an `MSAV` command that can be run from the DOS prompt to scan for and remove viruses.

Several other companies offer third-party virus protection software that can be configured to operate in various ways. If the computer is a standalone unit, it may be antiproductive to have the antivirus software run each time the system is booted up. It would be much more practical to have the program check floppy disks only, because this is the only possible entry way into the computer. A networked or online computer has more opportunity to contract a virus, however, because viruses can enter the unit over the network or through the modem. In these cases, setting the software to run at each bootup is more desirable.

Antivirus Programs

To run the MS-DOS antivirus program, type **msav.exe** in the command line. This should produce the antivirus program's main screen, as shown in Figure 13.18.

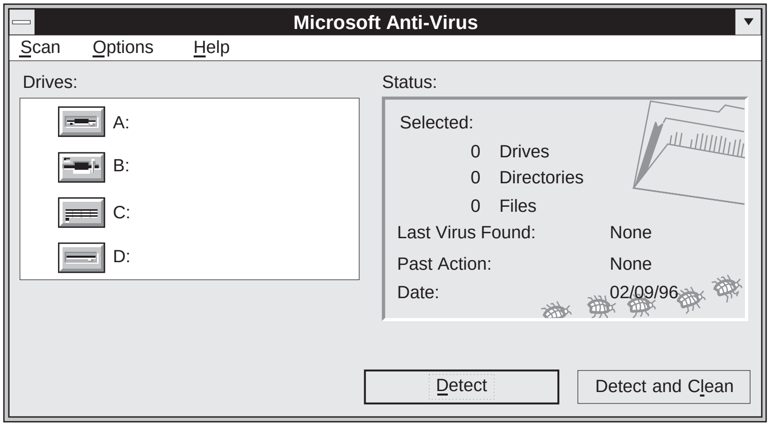
Complete the following steps:

1. Choose the Virus List from the Scan menu to produce the Virus List window. In this window, you can read the information about the viruses recognized by the current version of

the program. Because the list of viruses is long, you may search for virus names by typing the first few letters of the virus in the Search For box.

Figure 13.18

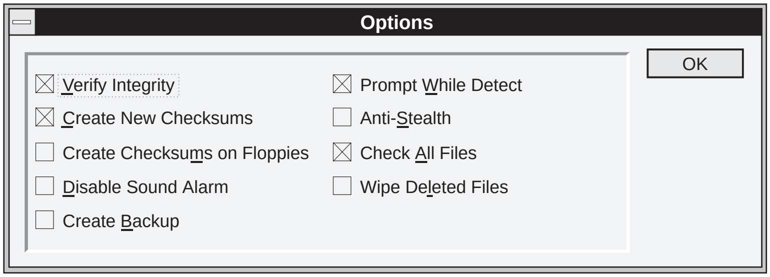
The antivirus screen.



2. Configure the antivirus program. Select Options to see the menu. Verify that an *x* appears in the box for Verify Integrity, Create New Checksums, Prompt While Detect, and Check All Files, as shown in Figure 13.19.

Figure 13.19

The Antivirus Options dialog box.



3. Select the Anti-Stealth and Wipe Deleted Files entries to activate them. An *x* should appear in the boxes beside them. Click on C: in the Drives: box. A small window will show that the program is reading the directories for the C: drive. The Status: window will show the number of drives, directories, and files it found on the system.

4. Run the antivirus program. Activate the Detect button to search for viruses. The program will first scan RAM memory for any viruses. If any are found in memory, they will be

cleaned from the system memory. The program will then search the files on the hard drive. Select the Update option if the program tells you that a file has been changed.

5. When the program finishes scanning the drive, a Statistics window appears. Exit the Statistics window, and select the Scan option to see the menu.
6. Alter the AUTOEXEC.BAT file to automatically start the antivirus program when the computer is booted up. Type **C:\DOS\VSAFE.COM** into the AUTOEXEC.BAT file. By doing so, the program will check for viruses when you Start a Program, Copy a File, or Save a File to disk. VSafe will notify you if any viruses infect your computer.

Floppy-Disk Drives

Objective

Unlike hard-disk drives, floppy drives are at least partially open to the atmosphere, and the disks may be handled on a regular basis. This opens the floppy-disk drive to a number of maintenance concerns not found in hard-disk drives. Also, the removable disks are subject to extremes in temperature, exposure to magnetic and electromagnetic fields, bending, and airborne particles that can lead to information loss.

Protecting Disks

Because the disk stores information in the form of magnetized spots on its surface, it is only natural that external magnetic fields will have an adverse effect on the stored data. Never bring disks near magnetic-field–producing devices such as CRT monitors, television sets, or power supplies. They should also never be placed on or near appliances such as refrigerators, freezers, vacuum cleaners, and other equipment containing motors. Any of these can alter the information stored on the disk.

Proper positioning of the drive and proper connection of peripheral interface cables helps to minimize noise and *radio frequency interference* (RFI). RFI can cause the drive to operate improperly. Magnetic fields generated by power supplies and monitors can interfere with the magnetic recording on the disk. The drive and signal cables should be positioned away from these magnetic-field sources. Magnets should never be brought near the drive unit.

Another major cause of floppy-disk failures is surface contamination. Several preventive measures will minimize disk contamination and lengthen the life expectancy of your disks. Although the disk is enclosed in a protective case or envelope whose liner sweeps contaminants from its surface, enough dust particles may collect to overpower the liner over time. Care should be taken to never touch the exposed surfaces of the disk. Store disks in their protective envelopes, and keep your computer area as clean and free from dust as possible.

There should be no smoking around the computer. Residues from tobacco smoke are a problem for floppy-disk drives because they tend to build up on the exposed surfaces of both the disks and the drive. These deposits are detrimental to both the drive and the disk because they gum up the close-tolerance mechanics of the drive and cause scratching to occur on the disk surface and the faces of the Read/Write heads. This makes the heads less effective in reading and writing information to and from the disk and eventually leads to failure of the disk and the drive.

The fact that the R/W heads ride directly on the disk surface produces a certain amount of contamination, and wear, on the disk and heads. During read and write operations, the abrasion between the heads and disk causes some of the oxide coating on the disk to be transferred to the head. This makes the head less effective in reading and writing operations, and eventually leads to the failure of the disk.

Additional measures to protect your disks include storing them in a cool, dry, clean environment, out of direct sunlight. Excessive temperature will cause the disk and its jacket to warp. Take care when inserting the disk into the drive so as not to damage its jacket or the drive's internal mechanisms.

Maintaining the Floppy Drive

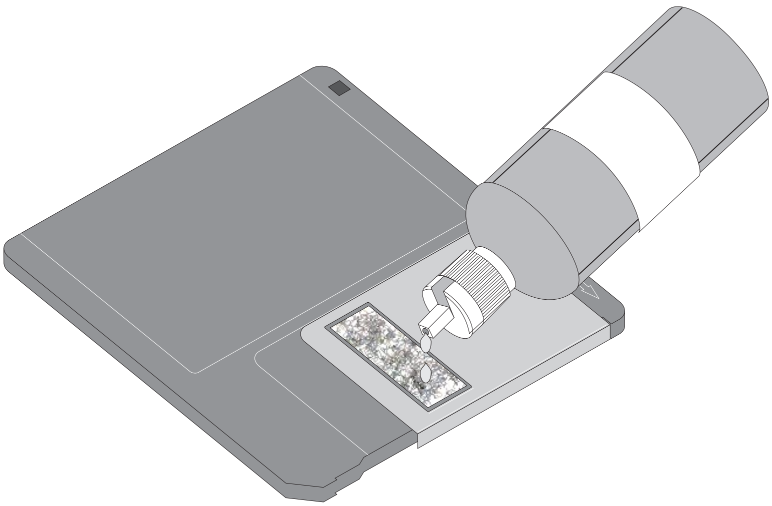
So far, each preventive action has involved the disk. There are, however, two procedures which the user can perform on the disk drive to ward off bigger maintenance problems. These are routine cleaning of the R/W heads (to remove oxide buildup) and periodic disk-drive speed tests and adjustments, when necessary.

Cleaning R/W heads removes residue and oxide buildups from the face of the head to ensure accurate transfer of data from the head to the disk. Two accepted methods may be used to clean the heads. These are special head-cleaning disks and manual cleaning of the heads. Head-cleaning disks are convenient to use, but some precautions must be taken when using them. There are, basically, two types of cleaning disks: dry (abrasive) disks, and wet (chemical) disks. Abrasive head-cleaning disks remove buildups as the disk spins in the drive. This is similar to using sandpaper to remove paint from a surface. These disks can be damaging to the head if used for too long at a time.

The dry disk must be left in the drive just long enough to remove the build-up on the head, but not long enough to scratch the head surface. Because of the difficulties of timing this operation, manufacturers have developed non-abrasive, cloth-covered disks which are used with a solvent solution. Depending on the type of kit you purchase, the disk may be premoistened or come with a separate solvent solution that must be applied to the disk before cleaning, as illustrated in Figure 13.20. The opportunity for abrasion of the head still exists with this type of cleaning disk. However, it is not as great as with the dry disks. The instructions that come with the cleaning kit should be consulted for proper usage and cleaning-time duration.

Figure 13.20

FDD cleaning disks.



A somewhat more complicated method of cleaning R/W heads is to clean them manually, as depicted in Figure 13.21. This operation involves removing the cover of the drive, gaining access to the R/W heads, and cleaning them manually with a swab that has been dipped in alcohol. Although this may appear to be a lot of work compared to the cleaning disk, manual cleaning is much safer for the drive. This is particularly true when combined with other cleaning, oiling, and inspection work. Together, these steps provide an excellent preventive-maintenance program that should ensure effective, long-term operation of the drive.

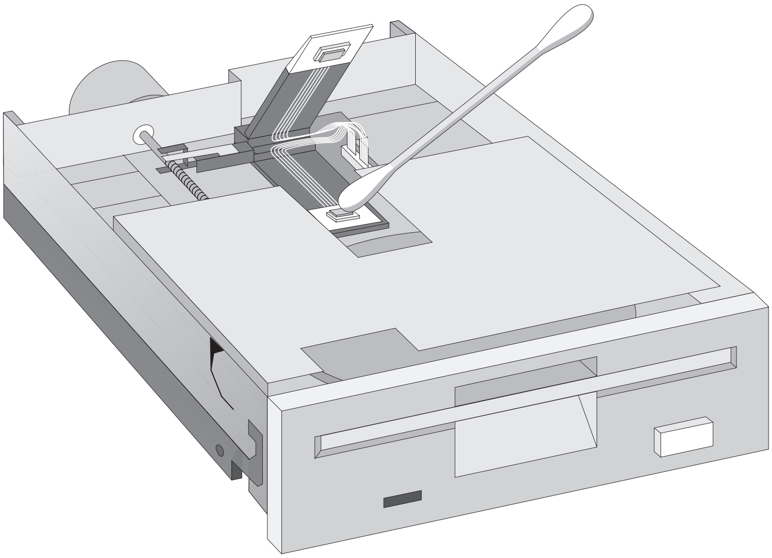


Figure 13.21

Cleaning the R/W heads.

The cleaning solution can be isopropyl alcohol, methanol, or some other solvent that does not leave a residue when it dries. Common cotton swabs are not recommended for use in manual cleaning, because they tend to shed fibers. These fibers can contaminate the drive and, in certain circumstances, damage the R/W heads. Instead, cellular foam swabs or lint-free cloths are recommended for manual head cleaning. Using either cleaning method, the interval of time between head cleanings depends on several factors, such as the relative cleanliness of your computer area and how often you use your disk drive. A good practice is to clean the heads after 40 hours of disk-drive operation. If read/write errors begin to appear before this time elapses, more frequent cleaning, or the use of higher-quality disks, may be required.

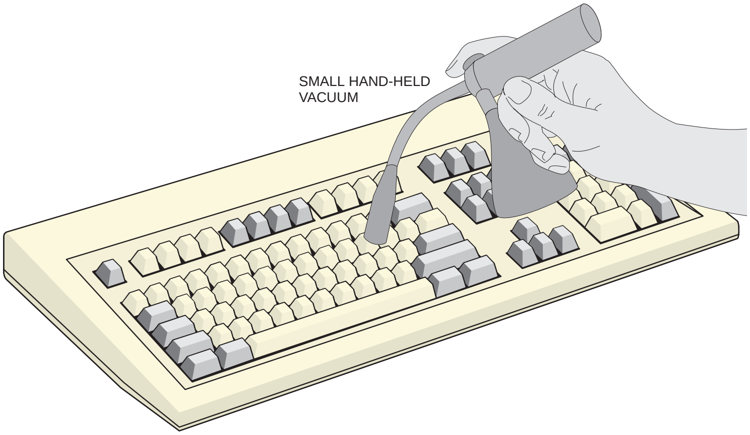
Input Devices

Input peripherals generally require very little in the way of preventive maintenance. An occasional dusting and cleaning should be all that's really required. Keep in mind, however, the following few common-sense items when using input devices. These "hints" should prevent damage to the device and ensure its longevity.

The keyboard should be vacuumed, as illustrated in Figure 13.22, when you are cleaning around your computer area. To remove dirt and dust particles from inside the keyboard, disassemble the keyboard and carefully brush particles away from the board with a soft brush. A lint-free swab can be used to clean between the keys. Take care not to snag any exposed parts with the brush or swab. To minimize dust collection in the keyboard, cover your keyboard when not in use.

Figure 13.22

Cleaning the keyboard.



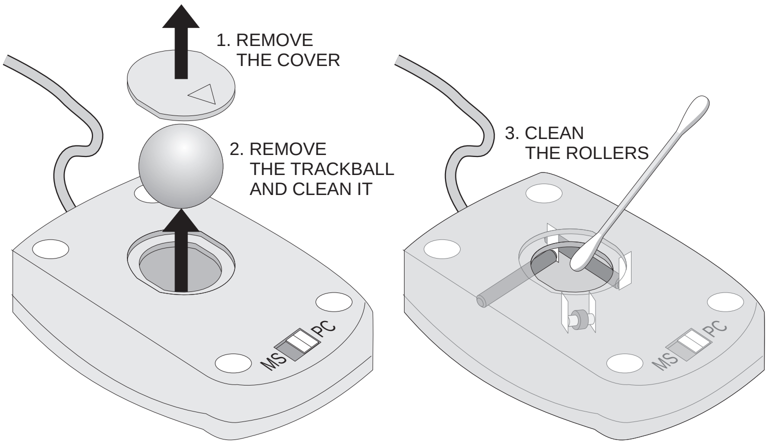
Never set keyboards or pointing devices on top of the monitor or near the edge of the desk where they may fall off. To prevent excessive wear on special keys, avoid applications and game programs that use keys in a repetitive manner. For these applications, use an appropriate pointing device such as a mouse or joystick for input.

When using a mouse, keep its workspace clear, dry, and free from dust. The trackball should be removed and cleaned periodically. Use a lint-free swab to clean the X and Y trackball rollers inside the mouse, as shown in Figure 13.23.

Never touch the lens of a light pen with your finger; the sensitivity of the pen will be diminished by the resulting smudge. As with detachable keyboards, keep the connecting cables of all pointing devices out of harm's way.

Figure 13.23

Cleaning the rollers in a Trackball mouse.



Printers

Objective

Because printers tend to be much more mechanical than other types of computer peripherals, they require more effort to maintain. Printers generate pollutants such as paper dust and ink droplets, even in everyday operation. These pollutants can build up on mechanical parts and cause them to wear. As the parts wear, the performance of the printer diminishes. Therefore, printers require periodic cleaning and adjustments to maintain good performance.

Dot-Matrix Printers

Adjust the printhead spacing, as described in Chapter 10, “Printers.” If the printhead is too far away from the platen, the print should appear washed-out. The tension on the printhead positioning belt should be checked periodically. If the belt is loose, the printer’s dot-positioning will become erratic. The belt should be reset for proper tension.

Cleaning the printer and its mechanisms periodically adds to its productivity by removing contaminants that cause wear. Vacuum the inside of the unit after applying antistatic solution to the vacuum’s hose tip. Wipe the outside with a damp cloth, also using antistatic solution. Brush any contaminant buildup from the printer’s mechanical components, using a soft-bristled brush. Never lubricate the platen assembly of the printer.

Clean the printer's roller surfaces. Use a damp, soft cloth to clean the surface of the platen. Rotate the platen through several revolutions. Do not use detergents or solvents on the rollers.

Use a non-fibrous swab dipped in alcohol to clean the face of the dot-matrix printhead. This should loosen up paper fibers and ink that may cause the print wires to stick. Apply a small amount of oil to the face of the printhead.

Clean the paper-handling motor's gear train. Use a swab to remove buildup from the teeth of the gear train. If the gear train has been lubricated before, apply a light oil to the gears, using a swab. Turn the platen to make sure the oil gets distributed throughout the gear train. Apply a light coating of oil to the rails that the head-positioning carriage rides on. Move the carriage assembly across the rails several times to spread the lubricant evenly.

To clean a dot-matrix printer, follow these steps:

1. Adjust the printhead spacing.
2. Check the tension on the printhead positioning belt.
3. Clean the printer and its mechanisms.
4. Clean the printer's roller surfaces.
5. Clean the surface of the platen.
6. Clean the surface of the dot-matrix printhead.
7. Clean the paper-handling motor's gear train.
8. Apply light oil to the gears, using a swab.
9. Turn the platen to distribute the oil.
10. Apply a light coating of oil to the rails.
11. Move the carriage assembly to distribute the oil.

Ink-Jet Printers

The spacing of the printheads in some ink-jet printers requires cleaning adjustment similar to those described for dot-matrix printers.

Clean the paper-handling motor's gear train. Use a swab to remove buildup from the teeth of the gear train. If the gear train has been lubricated before, apply a light oil to the gears, using a swab. Turn the platen to make sure the oil gets distributed throughout the gear train. Apply a light coating of oil to the rails that the printhead-positioning carriage rides on. Move the carriage assembly across the rails several times to spread the lubricant evenly.

To clean an ink-jet printer, follow these steps:

1. Adjust the printhead spacing.
2. Check the tension on the printhead positioning belt.
3. Clean the printer and its mechanisms.
4. Clean the printer's roller surfaces.
5. Clean the surface of the platen.
6. Clean the surface of the ink-jet printhead.
7. Clean the paper-handling motor's gear train.
8. Apply light oil to the gears, using a swab.
9. Turn the platen to distribute the oil.
10. Apply a light coating of oil to the rails.
11. Move the carriage assembly to distribute the oil.

Laser Printers

Use a vacuum cleaner to remove dust buildup and excess toner from the interior of the laser printer. Care should be taken to remove all excess toner from the unit. Vacuum the printer's ozone filter. Because water can mix with the toner particles in the printer, using wet sponges or towels to clean up toner inside the laser printer can create a bigger mess than the original one you were cleaning up. Remove the toner cartridge before vacuuming.

Use a damp cloth or denatured alcohol to clean the laser printer's rollers. Also, clean the paper-handling motor's gear train. Use a swab to remove buildup from the teeth of the gear train. If the gear train has been lubricated before, apply a light oil to the gears using a swab. Make sure the oil gets distributed throughout the gear train.

Clean the writing mechanism thoroughly. Use compressed air to blow out dust and paper particles that may collect on the lenses and shutters. If possible, wipe the laser lens with lint-free wipes to remove stains and fingerprints.

If accessible, use a swab dipped in alcohol to clean the corona wires. Rub the swab across the entire length of the wires. Take extra care to not break the strands that wrap around the corona. If these wires are broken, the printer will be rendered useless until new, monofilament wires can be reinstalled.

To clean a laser printer, follow these steps:

1. Remove dust buildup and excess toner from the interior.
2. Clean the laser printer's rollers.
3. Clean the paper-handling motor's gear train.
4. Apply light oil to the gears, using a swab.
5. Distribute the oil throughout the gear train.
6. Clean the corona wires.

In some laserprinter models, the toner cartridges are designed so that they can be refilled. At this time, the third-party refill cartridges are not typically as good as those from the manufacturer. However, they tend to be much cheaper than original equipment cartridges. If the output from the printer does not have to be very high quality, refilled toner cartridges might be an interesting topic to examine. To date, there are no regulations governing the disposal of laser-printer cartridges.

Preventive-Maintenance Schedule

Objective

There is no perfect PM schedule. However, the following is a reasonable schedule that can be used to effectively maintain most computer equipment. The schedule is written for use on a personal computer. From an outside-maintenance perspective, some of the steps will need to be shared with the daily user. As a matter of fact, most of the daily and weekly PM activities are carried out by the user.

Daily

Back up important data from the unit. This can be done to floppy disk, backup tape, another network drive, or some other backup media. Check computer ventilation to make sure that papers and other desk clutter are not cutting off air flow to the unit. Check for other sources of heat buildup around the computer and its peripherals. These sources include the following:

- Direct sunlight from an outside window
- Portable heaters
- Papers/books piled around the equipment

Weekly

Clean the outside of the computer and its peripheral equipment. Wipe the outsides of the equipment with a damp cloth. The cloth can be slightly soapy. Wipe dry with an antistatic cloth. Clean the display screen, using a damp cloth with the antistatic solution described earlier in this chapter. An antistatic spray can also be used for static buildup prevention.

Run `CHKDSK /f` on all hard drives to locate and remove any lost clusters from the drives. Run a current virus-check program to check for hard-drive infection. Back up any revised data files on the hard drive. Inspect the peripherals (mice, keyboard, and so forth) and clean them if needed.

Monthly

Clean the inside of the system. Use a long-nozzle vacuum cleaner attachment to remove dust from the inside of the unit. Wipe the nozzle with antistatic solution before vacuuming. A soft brush can also be used to remove dust from the system unit.

Clean the inside of the printer, using the same equipment and techniques as those used with the system unit. Check system connections for corrosion, pitting, or discoloration. Wipe the surface of any peripheral card's edge connectors with a lubricating oil to protect it from atmospheric contamination.

Vacuum the keyboard out. Clean the X and Y rollers in the trackball mouse, using a lint-free swab and a non-coating cleaning solution.

Use the Defrag utility to defragment the system's hard drive. Remove unnecessary TMP files from the hard drive. Check software and hardware manufacturers for product updates that can remove problems and improve system operation. Back up the entire hard-disk drive.

Six Months

Every six months, perform an extensive PM check. Apply an antistatic wash to the entire computer/peripheral work area. Wipe down books, desktop, and other work area surfaces with antistatic solution. Disconnect power and signal cables from the system's devices, and reseat them. Clean the inside of the printer. Run the printer's self-tests.

Use a software diagnostic package to check each section of the system. Run all system tests available, looking for any hint of pending problems.

Annually

Reformat the hard drive by backing up its contents and performing a high-level format. If the drive is an MFM, RLL, or ESDI drive, a low-level format should also be performed annually. Reinstall all the applications' software from original media, and reinstall all user files from the backup system. Check all floppy disks in the work area with a current antivirus program.

Clean the R/W heads in the floppy drive, using a lint-free swab. Cotton swabs have fibers that can hang up in the ceramic insert of the head and damage it. Perform the steps outlined under the monthly and semi-annual headings. While this is a good model PM schedule, it is not the definitive schedule. Before establishing a firm schedule there are several other points to take into consideration. These points include any Manufacturer's Guidelines for maintaining the equipment. Read the User's Guides of the various system components and work their suggested maintenance steps into the model.

Also, take note of the environment that the equipment is being used in and how heavy its usage is. If the system is located in a particularly dirty area, or an area given to other environmental extremes, it may need to be cleaned and serviced more frequently than if it were in a clean office environment. This is also true for systems that are subjected to heavy or continuous use. In an industrial environment, check with the management to see if there are any office or industry maintenance standards for servicing that are applicable.

Finally, use simple observation of the wear and tear on the equipment to gauge the need for additional or spot maintenance steps. Look for signs of extended use, such as frayed cords, missing back plates, keyboards with letters worn off, etc. to spot potential problems due to age or usage.

Over time, adjust the steps and frequency of the model to effectively cope with any environmental or usage variations. After all, the objective isn't to complete the schedule on time, its to keep the equipment running and profitable.

Safety Considerations

In most IBM compatibles, there are only two potentially dangerous areas. One of these is inside the display monitor, and the other is inside the power-supply unit. Both of these areas contain voltage levels that are lethal. However, both of these areas reside in self-contained units; you will normally not be required to open either unit.

As a matter of fact, you should never enter the interior of a CRT cabinet unless you have been trained specifically to work with this type of equipment. The tube itself is dangerous if accidentally cracked. In addition, extremely high-voltage levels may be present inside the CRT housing up to a year after electrical power has been removed from the unit.

Never open the power-supply unit either. Some portions of the circuitry inside the power supply carry extremely high-voltage levels.

Generally, there are no open-shock hazards present inside the system unit. However, you should not reach inside the computer while power is applied to the unit. Jewelry and other metallic objects do pose an electrical threat, even with the relatively low voltage present in the system unit.

Never have liquids around energized electrical equipment. It is a good idea to keep food and drinks away from the computer equipment at all times. When cleaning around the computer with liquids, make certain to unplug all power connections to the system and its peripherals beforehand. When cleaning external computer cabinets with liquid cleaners, take care to prevent any of the solution from dripping or spilling into the equipment.

Do not defeat the safety feature of three-prong power plugs by using two-prong adapters. Periodically examine the power cords of the computer and peripherals for cracked or damaged insulation. Replace worn or damaged power cords promptly. Never allow anything to rest on a power cord. Run power cords and connecting cables safely out of the way so that they don't become "trip" or "catch" hazards. Remove all power cords associated with the computer and its peripherals from the power outlet during thunder or lightning storms.

Don't apply liquid or aerosol cleaners directly to computer equipment. Spray cleaners on a cloth, and then apply the cloth to the equipment. Freon-propelled sprays should not be used on computer equipment; these sprays can produce destructive electrostatic charges.

Check equipment vents to see that they are clear and have ample free-air space to allow heat to escape from the cabinet. Never block these vents, and never insert or drop objects into them.

Objective

Laser printers contain many hazardous areas. The laser light can be very damaging to the human eye. In addition, there are multiple high-voltage areas in the typical laser printer and a high-temperature area to contend with as well. The technician is normally protected from these areas by interlock switches built in to the unit. However, it is often necessary to bypass these interlocks to isolate problems. When doing so, proper precautions must be observed—such as avoiding the laser light, being aware of the high temperatures in the fuser area, and taking proper precautions with the high-voltage areas of the unit.

This covers the major safety precautions and other considerations that you need to be aware of while working on computer equipment. Most of all, use common sense and sound safety practices around all electronic equipment.

Disposal Procedures

As with any mechanical device, a computer eventually becomes obsolete in the application that it was originally intended for. Newer machines, with improved features, arise to replace earlier models. And slowly but surely, components fail and get replaced. Then comes the question: What to do with the old stuff? Can it just be placed in the garbage bin so that it is hauled to the landfill and buried?

In today's world of environmental consciousness, you might not think so. Computers and peripherals contain some environmentally unfriendly materials.

Most computer components contain some level of hazardous substances. Printed circuit boards consist of plastics, precious metals, fiberglass, arsenic, silicon, gallium, and lead. CRTs contain glass, metal, plastics, lead, barium, and rare earth metals. Batteries from portable systems can contain lead, cadmium, lithium, alkaline manganese, and mercury. Although all these materials can be classified as hazardous materials, so far there are no restrictions when it comes to placing them in the landfill.

As mentioned in an earlier section, laser-printer toner cartridges can be refilled and recycled. This should only be done, however, in draft-mode operations where very good resolution is not required. Ink cartridges from ink-jet printers can also be refilled and reused. Like laser cartridges, they can be very messy to refill and often do not function as well as new cartridges do. In many cases, the manufacturer of the product will have a policy of accepting spent cartridges.

For both batteries and cartridges, the desired method of disposal is recycling. It should not be too difficult to find a drop site that will handle the recycling of these products. On the other hand, even Non-hazardous Sub-title D dump sites can handle the hardware components if need be.

Fortunately, several charitable organizations around the country take in old computer systems and refurbish them for various applications. Contact your local Chamber of Commerce for information about such organizations. The Internet also has several computer-disposal organizations that will take old units and redistribute them. In addition, a few companies will dispose of your old computer components in an “environmentally friendly” manner—for a fee.

In addition to the computer parts that contain hazardous materials, many of the cleaning substances used on computer equipment can be classified as hazardous materials. When it comes to disposing of the chemical solvents used to clean computers, as well as the containers they come in, it will normally be necessary to clear these items with the local waste-management agencies before disposing of them. Many dump sites will not handle free liquids. Therefore solvents and other liquid cleaning materials must be properly categorized and disposed of at an appropriate type of disposal center.

All Hazardous materials are required to have *Material Safety Data Sheets* (MSDS) that accompany them when they change hands. They are also required to be on hand in areas where hazardous materials are stored and commonly used. The MSDS contains information about:

- What the material is
- Its hazardous ingredients
- Its physical properties
- Fire and explosion data
- Reactivity data
- Spill or leak procedures
- Health hazard information
- Any special protection information
- Any special precaution information

This information sheet is required to be provided by the supplier of the hazardous material. If you supply this material to a third party, you must also supply the MSDS for the material. The real reason for the sheets is to inform workers and management about hazards associated with the product and how to handle them safely. It also provides instructions about what to do if an accident occurs involving the material. For this reason, employees should know where the MSDS are stored in their work area.

Summary

The focus of this chapter has been to present important points for inclusion in the preventive-maintenance programs associated with personal computer systems. The first section of the chapter dealt with typical cleaning chores. The following section discussed the danger and causes of electrostatic discharges and provided information about how to eliminate them. An additional section focused on the problems that revolve around fluctuations in the computer's incoming power line. Different types of uninterruptible power supplies (UPS) were discussed, along with other power-line conditioning devices.

The midsection of the chapter featured preventive-maintenance procedures for the system's different components. Important HDD support utilities such as backup, defragmentation, and antivirus protection were spotlighted here. A suggested PM schedule was also presented. This is time-proven information and should always be shared freely with customers.

The final section of the text involved safety issues concerning computer systems. Although not an intrinsically unsafe environment, some areas of a computer system can be harmful if approached without caution.

At this point, review the objectives listed at the beginning of the chapter to be certain that you understand the information associated with each one and that you can perform each item listed there. Afterward, answer the review questions that follow to verify your knowledge of the information.

Lab Exercises

There are hands-on lab procedures that correspond to the theory materials presented in this chapter. Refer to the Lab Manual and perform Procedures 35 - DOS Antivirus, 36 - HDD Utilities, and 37 - Logical Versus Physical Drives.

Review Questions

1. List the two most dangerous areas of a typical microcomputer system, and describe why they are so dangerous.
2. What is the most common cause of ESD in microcomputer systems?
3. Name three devices used to minimize ESD in the repair area.
4. Do viruses normally attack the system's CMOS settings?
5. The best general-purpose cleaning tool for computer equipment is _______.
6. List three type of backups that can be performed with most backup utilities.
7. Describe common PM procedures for laser printers.
8. How is excess toner normally removed from a laser printer?
9. Are there any restrictions on disposing of a spent toner cartridge?
10. List three types of computer viruses, and describe how they are different.
11. How are most computer viruses spread from computer to computer?
12. Why are defragmentation programs run on computers?
13. What is the purpose of running a `CHKDSK` operation before performing a backup or defrag operation on the hard drive?

14. If the DOS defrag option is not available, what other DOS command could be used to eliminate file chaining on a hard-disk drive?

15. The best method of protecting computer equipment from a thunderstorm is to _______.

16. ESD is most likely to occur during periods of _________ humidity.

17. The best method for transporting electronic devices is _________.

18. What is the best method of cleaning a keyboard?

19. Describe the disadvantage of using the DOS or Windows backup utility to perform a total backup of a large hard disk drive.

20. What type of backup device is typically used to store large banks of information?

21. How often should the platen assembly of a dot-matrix printer be lubricated?

22. List computer-related PM items that should be performed annually.

23. Name two characteristics that should be checked carefully before purchasing a UPS for a given computer system.

24. Describe the normal duration of a voltage spike.

25. Define a voltage sag.

Review Answers

1. The inside of the monitor and the inside of the power supply. Both units house potentially dangerous voltage levels inside their housings. For more information, see the section titled “Safety Considerations.”

2. Moving people. For more information, see the section titled “Electrostatic Discharge.”

3. An antistatic wrist strap, rubber antistatic mats, and a humidifier. For more information, see the section titled "MOS Handling Techniques."
4. No. Most computer viruses affect the boot sector, executable files, and legitimate programs. For more information, see the section titled "Antivirus Programs."
5. A damp cloth. For more information, see the section titled "Cleaning."
6. Total, selective, and differential backups. For more information, see the section titled "Backup Types."
7. Clean dust and stray toner from the inside of the unit. Clean the rollers, gear train, and corona wires. Lubricate the gear train. For more information, see the section titled "Laser Printers."
8. It is best to vacuum toner out of the printer. For more information, see the section titled "Laser Printers."
9. No, not currently. However, it is economical to recycle toner cartridges unless the output from the printer has to be of very high quality. For more information, see the section titled "Laser Printers."
10. Boot sector viruses, file infectors, and trojan horses. These viruses differ in how they infect the system. For more information, see the section titled "Antivirus Programs."
11. By infected disks exchanged between individuals. For more information, see the section titled "Antivirus Programs."
12. To rearrange information on the disk in a logical sequence so that it is quicker to read and write. For more information, see the section titled "HDD Defragmentation."
13. To locate and remove lost allocation units (clusters) that slow down the operation of the drive. For more information, see the section titled "`CHKDSK`."

14. The CHKDSK or SCANDISK utilities. For more information, see the sections titled "CHKDSK" and "SCANDISK."

15. Unplug it and all of its peripherals from the power outlet so that there is no path for lightening to follow. For more information, see the section titled "Grounds."

16. Low. For more information, see the section titled "Electrostatic Discharge."

17. To place them in an antistatic bag. For more information, see the section titled "MOS Handling Techniques."

18. Vacuuming or blowing dust out and cleaning with a damp cloth. For more information, see the section titled "Input Devices."

19. A large number of disks are required to back up the contents of a typical hard-disk drive. For more information, see the section titled "Hard-Disk Drives."

20. A tape drive. For more information, see the sections titled "Hard-Disk Drives" and "Other Backup Methods."

21. Never. For more information, see the section titled "Dot-Matrix Printers."

22. Reformat the hard drive, reinstall all the applications software, check all floppy disks, clean the R/W heads in the floppy drive, and perform the steps outlined in the monthly and semi-annual sections. For more information, see the section titled "Annually."

23. Ampere-hour rating and wattage rating. For more information, see the section titled "Uninterruptible Power Supplies."

24. Nanoseconds. For more information, see the section titled "Power Line Protection."

25. A voltage sag is an under-voltage condition that normally lasts for a few milliseconds. For more information, see the section titled "Power Line Protection."

Appendix A

Customer Satisfaction

Customer Service Skills

The object of this appendix is to discuss customer-service skills that employers find desirable in their employees. Many companies have formal customer-service guidelines that they make all employees aware of. However, these guidelines are not universal.

Everything discussed so far in this text has concerned the development of good technical skills. For the most part, a high level of technical proficiency alone is not enough to sustain a career in the world of computer service. For most of the service jobs available, good customer skills are just as important as good technical skills. In most cases, they are equal partners for a successful career.

The field, or bench, technician must possess a good set of interpersonal skills to be able to handle customers. It has been said that we don't fix computers; we fix customers. How we are perceived is as important as how well we perform. In the end, it is customer satisfaction (with the product, the service, and the supplier) that creates a successful business and continued employment.

Customer-service skills are generally referred to as `soft skills`, because they cannot be tested easily with a written or hands-on test. They are skills that can be learned and practiced, however. The following paragraphs contain key points to consider in the area of providing customer service.

Prepare

Review customer history before contacting them or going to their sites to perform work. In particular, see whether the problem you are going to work on is a repetitive problem or a new occurrence. Check the urgency of the call and the customer's priority level.

Research the type of equipment the call concerns. Determine whether any special tools are needed or any special parts may be involved in the repair. Make sure your documentation is in order. It may be necessary to check the customer's status with your company. Make sure that you have all the manuals, replacement parts, and tools you will likely need. Do you have the needed ESD equipment, meters, hand tools, and so forth? Make sure the tools and parts are in good working order. Set realistic schedules. Make appointments that you can keep. Always notify customers as soon as possible about any appointment changes, service delays, complications, or setbacks that may occur. These things happen to everyone, and your best defenses against customer dissatisfaction are promptness and good communication.

Establish Rapport

A good practice is to learn your customers by name and greet them personally. Collect business cards and include copies in customer folders—have them in your pocket during the call. Always deal with people as individuals, not by stereotypes such as order/entry person, receptionist, manager, and so forth. Be as open, friendly, and approachable as your personality will allow. This is an area that most of us can always work on. Politeness is a valuable quality to possess. However, it should never be forced, contrived, or overdone. Your greatest weapon in this area is your expertise. Avoid politically, or racially sensitive topics. These have no place in business settings.

Establish Your Presence

Make eye contact when you speak to customers. Maintain alert body posture and calm facial expressions. Keep your voice calm when perplexing situations arise. Your presence can be used to control the excitement level of the customer. If you appear calm, collected, and confident, the customer probably won't get too excited either. Avoid moodiness in the workplace. This can undermine your credibility with the customer. You begin each new customer contact with a 100% rating. In the end, you will be left with whatever points you have not given away. Doing an efficient, professional, and complete job does more to ingratiate you to customers than almost anything else you can do. If customers get to the place where they are sure that you will always be able to solve their problems, you have the best rapport that could be achieved in a business setting.

Be Proactive

Provide a customer with a preventive-maintenance plan and explain how it contributes to the continued productivity of the customer's equipment. Take time to illustrate proper methods of handling consumables and to identify items to be on the watch for. Show your customers how to install and change printer ribbons, ink cartridges, toner cartridges, and so on. Demonstrate the use of virus-protection products and backup utilities. Advise them of potential environmental hazards such as disposing of toner cartridges.

Scenario 1

A customer calls with a RAM problem that has been common in a particular model of computer your company sells. What should you tell the customer about the product?

A. Oh, yeah, we have that happening all over. We'll be getting everyone fixed up real soon.

B. We've had a whole batch of bad RAM.

C. Troubleshoot the problem as you would any other phone support problem.

D. Offer to send the customer replacement RAM.

Alert customers of potential system problems or productivity-related issues concerning their systems. Identify noisy system components that may need care in the near future. Suggest system changes that could improve performance and explain how this is possible. Keep customers aware of service bulletins and advisories concerning their equipment.

Research customer requests for recommendations and advise about future directions and equipment. This option should normally be taken only if requested by the customer, however. Also be sensitive to the level of the person to whom you are making recommendations. There are workers who have never met a new piece of hardware or software they didn't like. However, their superiors may be quite happy without the production/downtime/cost trade-offs that changing could bring. Don't incite the customer's employees with the latest and greatest product if their management hasn't signaled the way.

Listen and communicate. One of the attributes that makes a good customer-service or repair person is the ability to `actively listen` to the customer. Real listening means not just hearing what the customer has to say, but trying to pin down what he or she means. Mentally (and maybe physically) identify key points as the customer describes the nature of the problem. Don't interrupt customer descriptions before you have all the details. Even if you are sure that you know what is going on after the first sentence, have patience to listen to the complete description. This is not only common courtesy, but also serves to uncover extra data about the problem.

Scenario 2

You are sent to set up a desktop-publishing computer for a publisher, and you discover that they are using a publishing program that you know does not have all the features of a competing program. You are sure that they could be much more productive using the other program. How do you convey this to the customer?

A. You don't, because they are installing a new program and do not need to be told they don't have the best.

B. Get the supervisor alone, and recommend the other program in private.

C. Mail an advertisement of the better product to the manager.

D. Tell the operators about the features of the other product in confidence so that they will know what to look for the next time.

Pay attention to the customer's body language and other nonverbal clues. Pay attention to body posture, hand gestures, facial cues, and voice inflection to gauge anger, franticness, and so on.

Listening is also a good way to eliminate the user as a possible cause of the problems occurring. Many cartoons have been created in service newsletters about the strangest user-related calls ever received. Part of your job is to determine whether the user could be the source of the problem—either trying to do things with the system that it cannot do, or not understanding how some part of it is supposed to work. If you find this to be the case, work with the user to clarify the realistic uses of the system. This is a point where it may be appropriate to suggest advanced training options. However, such suggestions should be made discreetly.

Scenario 3

You arrive on a service call and the office supervisor turns on the malfunctioning machine. She begins to explain what she thinks the problem is, but you can tell from the operation of the machine that it is something else. What course of action should you pursue?

A. Listen to the explanation until she is finished, and then fix the machine.

B. Sit down and fix the machine while she is describing the rest of the problem to you.

C. Begin troubleshooting the problem she is describing until she leaves, and then fix the problem.

D. Stop the explanation and tell her that you are pretty sure that you already know what the problem is.

The ability to `communicate clearly` is the other trait most looked for in service people. Allow customers to talk through their problems. Use probing questions for clarification purposes and to make sure you understand what the user is describing at each step. In doing so, they may come up with clues they haven't thought of before. Help them to think through the problem by asking them organized questions. With equipment down, the customer may be under some stress and might not be thinking as rationally as possible. Choose words and questions that do not put the customer or their employees in a bad light. (For example: What have you done now? is not likely to set the proper tone with a customer who has more problems than he needs at the moment.) Adjust the pace and flow of your conversation to accommodate the customer.

Avoid quick analysis statements. Repeatedly changing your position kills customer confidence. Also, avoid or minimize surprises that pop up (such as unexpected charges or time requirements). Try to manage the customer's expectations by being as up front as possible about what you can accomplish and the scope of services you can provide. If the customer has a networking problem and you are the computer repair person, the customer should not be allowed to believe that you are going to get everything working before you leave (unless the network falls within the scope of your normal work).

If the person you are working with in a company is the MIS person, network administrator, or engineer, take her lead and follow her instructions. Avoid situations of who knows more. Try to quickly recognize the technical abilities of the people you are working with. Adjust your conversation to accommodate them. For technically challenged customers, avoid jargon. It will be confusing to them and cause customer dissatisfaction with you, even if you do a great job. Clarify your terminology with such people, and be careful to avoid `talking down` to them or `patronizing` them. On the other hand, if the customer is technically literate, be careful not to insult her intelligence by over-explaining things to her. In this case, use technical terms as appropriate, and use them correctly. Watch for signs of misunderstanding, and explain things in greater detail as necessary.

Scenario 4

While working with a relatively inexperienced customer over the telephone, you become aware that the customer is having great difficulty following your directions. How can you help the customer even though he cannot see you?

A. Send the customer fax drawings of steps you need him to perform.

B. Ask the customer whether you can talk to someone else, to get a fresh perspective on the problem.

C. Check your conversation and try to communicate more clearly.

D. Ask the customer to fax you drawings of what he is experiencing.

Follow up on unresolved issues. For incomplete calls such as those requiring additional parts, assess the customer's need and restore as much functionality to the system as possible or needed. Clean up and organize parts removed from the system so that they will not be in the way or be removed before your return to the site. Keep the customer informed about progress of unresolved issues, such as when parts are expected. If problems are intermittent, set up a schedule and procedure to work with the customer to pin down circumstances that cause the problem to recur.

Be Responsive. Concentrate on the customer's problem or request. Give preeminence to the customer's sense of urgency. Relegate paperwork and administrative duties to a secondary level until the customer's problems have been fully aired. Don't undermine the customer's sense of urgency. Work with their priorities.

Schedule steps to fulfill any unresolved problems to show commitment to getting the customer's problems solved. In this way, the customer will be assured that they are not being left astray. Don't multitask while working directly with a customer. Focus on the task at hand, and keep it in the forefront. Avoid distractions in the customer's presence. Act on the customer's complaints.

Be Accountable. Document your promises and dates so that you may demonstrate accountability to your customers. Follow up on return dates for yourself and/or equipment. Take personal responsibility for being the single point for the service call—contacting specialists, dealing with parts vendors, and so forth. It is your show, so run it.

Scenario 5

An irate customer calls, complaining that a technician from your company has recently performed a software upgrade on their system and now the modem will not connect with other modems. How should you handle the customer?

A. I'm sure none of our technicians would have left a condition like that. Let's see what the problem is.

B. Give me the technician's name, and I'll have him get back to you as soon as he returns to the office.

C. Please describe the symptoms to me, so I can see what might be causing the problem.

D. This is really easy. Take the top off of the computer, and check to see that the card is installed securely.

Be Flexible. If a problem runs beyond your abilities, take the initiative to move it to the next level of authority. Never leave a customer hanging without a path to get his problems settled. Provide alternatives to the customer when possible (downtime scheduling, loaner equipment availability, and so forth).

Be Professional. You should always make certain that your attire is clean, neat, and appropriate. You should establish a good rapport with your customers, but you should always maintain a professional distance from them. You cannot afford to be their support or confidant in dealings with the company that you work for. The apparent opportunity to gain the inside track with the customer, at your company's expense, cannot work out in your favor. Remember that the customer sees you as an extension of that company. Therefore, you can only be as good as your company is to the customer. If my computer is broken, I don't need a friend; I need the best repair person I can find.

Listening and communications skills are equally valuable when performing `phone support`. Because you cannot see the customer or interpret his body language, it is even more important to assess the state of the customer quickly. It is also important to determine the technical abilities of the user as quickly as possible. Asking a receptionist to remove the cover of a computer is not normally an accepted practice.

When giving instructions over the phone, be precise. Provide detailed instructions for work to be done, and ask a lot of questions about what is happening on the other end. Cellular or cordless phones are extremely valuable tools for customers using phone support. At your end of the phone, take good notes of what the user has been instructed to do so that you can review those notes as needed.

It is much more beneficial to establish the relationship with the customer based on your abilities and integrity. Have something good to say about the customer's facilities, if possible. If not, don't comment on them at all. The same goes for equipment that they have chosen. They aren't paying you for consulting services, and disparaging comments about their choices won't win you any points.

From time to time, you may be exposed to customer information that is of a sensitive nature. Respect the confidentiality of this information. Never reveal financial information that you have obtained from a customer's system. This includes to friends and, especially, employees of the customer.

Scenario 6

You receive a call from a customer who buys thousands of dollars of computer equipment from your company each year. The customer cannot get your company's computer to work with a printer that was purchased through another supplier. She wants you to get the system running. How should you react?

A. I'm really sorry, but we can't work on equipment purchased from another vendor.

B. Sure, how can I help you?

C. Let me clear this with my supervisor.

D. You'll have to tell my supervisor that you want me to do this. I'm sure that it will be all right, because you do so much business with us.

Avoid distracting employees while you are working at a customer's site. Work as unobtrusively as possible. Ask permission to use the customer's facilities, such as the telephone, copier, or other equipment. For example, if paper is required to test a printer you are repairing, ask an appropriate person for it. Don't just get it for yourself. Straighten up the work area before leaving it (don't leave the paper from the print tests lying around, for example).

Never break copyright regulations by loading or giving away illegal software. One of the leading causes of computer virus infection is pirated software. Not only do you run this risk in giving away copies, it's illegal and can get you introduced to various people you never really wanted to meet—lawyers and judges. Not only that, it could cost you your job.

Handle conflicts appropriately. Inevitably, you will run into a customer who is having a bad day. No matter what you do, you will not be able to keep him from getting angry over his situation. What you can do is realize that this is the case and attempt to de-escalate the situation. The easiest thing to do is let the customer vent without replying. Try to avoid taking a defensive stance;

this signals a conflict point. Redirect the conversation to creating solutions to the problems.

Remain calm, talk in a steady voice, and avoid making inflammatory comments. As soon as possible, withdraw from the confrontation and let the situation cool off. Inform your superiors as quickly as possible so that you have inside support and so that a plan of relief can be implemented.

Handle paperwork, and follow up. A number of nontroubleshooting, nonrepair activities must be handled in order to have an efficient organization. As a service person, you are not alone. You are part of a team that must communicate effectively for the system to operate smoothly. Other members of the team won't be able to do their jobs effectively if you don't follow through on yours. Process paperwork as soon as possible so that it can move through your system.

Scenario 7

A customer who has picked up a repaired computer from your store brings it back within a few hours, complaining that it doesn't work. What should you say?

A. What happened to it?

B. Sometimes I can't believe our technicians can find their way home at night. I'll get this thing fixed up for you.

C. It was working when it left here. I don't know what could have happened to it. Let's take a look at it.

D. Did it ever work when you got it home?

Follow up with people you have delegated tasks to, to make sure that those tasks are being taken care of. Use an organizational aid to coordinate jobs, appointments, and activities with coworkers. Prioritize commitments and resist the urge to overbook appointments. If it takes half an hour to get across town, never schedule an appointment there for half an hour after you should be finished at the first site. Maintain time allotments established for the completion of different tasks.

Maintain an orderly work area. Handle jobs one at a time so that components from one job do not get mixed up with components from another job. Store equipment not being used so that there is ample room to work and so that these items do not become a safety hazard.

Keep an inventory of parts and equipment in your area of responsibility. Order parts needed for a job promptly, and keep a log of when they should arrive. If the parts do not arrive as scheduled, you should have a reminder that they are still missing. This reminder enables you to track repair parts so that when a customer calls to check on her equipment or problem, you will have the information at hand.

Tag parts brought into the work area so that they do not get lost or mishandled. Store them in a way to limit potential damage by environmental factors such as electrostatic discharge. Keep good records—include all pertinent information about the part, the problem description, repair notes, and customer name and location.

Appendix B

Glossary

A

A: drive

The commonly understood term designating the first floppy-disk drive in Microsoft's DOS microcomputer operating system.

access time

1) When specifying memory access, the amount of time required to transfer a character to/from the MPU to/from RAM. This speed is normally given in nanoseconds.

2) When specifying disk access, the average time required to position the read/write head over a specific track. This speed is normally given in milliseconds.

accumulator

A special-purpose register that is used by the MPU to store the results of ALU operations. It also serves as the source of one of the operands for most ALU operations.

ACK (ACKnowledge)

A data communications code used by the receiver to tell the transmitter it is ready to accept data. During a data transfer this signal is continually used to indicate successful receipt of the last data character or block and to request more.

adapter

A device that permits one system to work with and connect to another. Many I/O device adapters interface with the microcomputer by plugging into the expansion slots on the system board. These specialized circuit boards are often called adapter cards.

ADB (Apple Desktop Bus)

A serial interface that was designed for connecting mice, keyboards, and other input devices to the MAC. The interface acts in a pass-through manner so that the input devices are chained together on a single port.

ADC (Analog-to-Digital Converter)

A device that converts a continuously varying signal (analog) into a binary coded signal (digital) for use by a computer.

address

The unique location number of a particular memory storage area, such as a byte of primary memory, a sector of disk memory, or a sector of a peripheral device itself.

address bus

A unidirectional pathway that carries address data generated by the MPU to the various memory and I/O elements of the computer. The size of this bus determines the amount of memory a particular computer can use and therefore is a direct indication of the computer's power.

alphanumeric data

Data such as names and addresses, as distinguished from numeric data such as monetary and quantity figures.

ALU (Arithmetic Logic Unit)

The subsection of the CPU where the actual math and logic operations are performed, two binary numbers at a time.

AM (Amplitude Modulation)

A method of data transmission that modulates the data into a carrier by varying the carrier amplitude.

analog monitor

A video display that is designed for use with a variety of computer or television standards, including the common NTSC composite video standard. Analog monitors offer a high-resolution picture. Some models are capable of many different sweep frequencies (multisync), and most accept separate red, green, and blue (RGB) signals for sharper contrast than digital monitors.

AppleTalk

The Apple protocol for networking.

ASCII (American Standard Code for Information Interchange)

The 7-bit binary data code used in all personal computers, many minicomputers, and also in communications services. Of the 128 possible character combinations, the first 32 are used for printing and transmission control. Because of the 8-bit byte used in digital computers, the extra bit can be used either for parity checking or for the extended ASCII set of characters, which includes foreign language characters and line-draw graphic symbols.

ASIC (Application-Specific Integrated Circuit)

An ASIC is a specialized, very large-scale IC that is created to include large blocks of circuitry that have become standardized. For example, a Multi I/O ASIC combines the circuitry that controls the HDD/FDD/serial ports/printer port/game port into a single IC.

asynchronous transmission

A method of serial data transmission where the receiving system is not synchronized, by a common clock signal, with the transmitting system.

AT bus

Also referred to as ISA (Industry Standard Architecture Bus). The 16-bit data bus introduced in the AT-class personal computer.

attribute byte

A data byte specifying the display characteristics of a single character of text such as highlight, underline, reverse video, or blinking.

attribute controller

The portion of the VGA video controller that converts data from the video memory and formats it for display on the video display.

auto dial

A modem feature that automatically accesses the line and dials the telephone of the answering modem in order to establish a connection.

B

B: drive

The commonly understood term designating the second floppy-disk drive in the DOS and OS/2 microcomputer operating systems.

BASIC (Beginners All-purpose Symbolic Instruction Code)

A high-level programming language originally developed in the mid-1960s for timesharing mainframe use. It now has become widely used on all types of computers.

BAT file (BATch file)

A file extension name used to identify a batch file in Microsoft DOS versions. A batch file, created by a word processor, contains a list of DOS commands that are executed as if each was typed and entered one at a time.

baud rate

The number of electrical state changes per second on a data communication line. At lower speeds, the baud rate and the bits-per-second rate are identical. At higher speeds, the baud rate is some fraction of the bits-per-second rate.

BCD (Binary Coded Decimal)

A binary coding system in which individual decimal numbers 0–9 (digits) are converted into separate groups of four binary bits, one group for each decimal digit.

binary

This means two. In conjunction with digital computers, all data is processed only after being converted into binary numbers consisting of the two digits 0 and 1.

BIOS (Basic Input Output System)

See *ROM BIOS*.

bit (binary digit)

One digit of a binary number (0 or 1). Groups of bits are manipulated together by a computer into various storage units called nibbles, bytes, words, or characters.

bit map

A term used in computer graphic to describe a memory area containing a video image. On a monochrome screen, one bit in the map represents one pixel; in color or gray scale monitors, several bits in the map may represent one pixel.

bit stream

A continuous flow or transfer of binary signals.

boot

To start the computer. It refers to the word *bootstrap*, because the straps help in pulling boots on, just as the bootable disk helps the computer to get its first instructions.

bootable disk

Normally refers to a floppy disk containing the computer operating system.

bps (bits per second)

A term used to measure the speed of data being transferred in a communications system.

buffer

1) Also known as bus drivers; a special class of digital amplifying devices used to match dissimilar circuit characteristics of various IC devices, and to increase the number of device inputs that an output can successfully drive. These devices usually are equipped with tri-state capabilities.

2) A class of memory registers and devices used to match data transfer speeds between the computer and a peripheral. Can be as simple as a single register or as complex as a large portion of RAM storage being used to accommodate large data transfers such as DMA operations.

bulk storage

Refers to any data storage medium (tape or disk) not used for high-speed execution.

bus

A parallel collection of conductors that carry data or control signals from one unit to another.

bus master

Any class of MPU having the capability to take control of the system buses of a computer.

bus mouse

A type of mouse that plugs into an adapter card rather than a serial port.

byte

The most common word size used by digital computers. It is an 8-bit pattern consisting of both a high- and a low-order nibble. Computers of any size are frequently described in terms of how many bytes of data can be manipulated in one operation or cycle.

C

C: drive

The commonly understood term designating the system or first hard-disk drive in the DOS and OS/2 microcomputer operating systems.

cache

An area of high-speed memory reserved for improving system performance. Blocks of often-used data are copied into the cache area to permit faster access times. A disk cache memory area is actually located on board the disk controller card.

cache controller

An MPU with the specific task of managing a sophisticated cache memory system.

carriage

The part in a printer or typewriter that handles the feeding of the paper forms.

cartridge

A removable data storage module, containing disks, magnetic tape, or memory chips, and inserted into the slots of disk drives, printers, or computers.

CCP (Chip Carrier Package)

A very popular IC package style which, unlike the DIP, has pins around its entire circumference. A CCP takes up to 50% less surface area on a printed circuit board than an identical circuit in DIP form. Pins are “gull-wing” shaped and set on solder pads of the PCB. These chips are more difficult to test and replace than DIPs.

Centronics interface

The 36-pin standard for interfacing parallel printers, and other devices, to a computer. The plug, socket, and signals are defined.

CGA (Color Graphics Adapter)

A low-resolution, IBM video standard for text and graphics, requiring the use of a digital RGB Color Display monitor. It has been superseded by the EGA, VGA, and SuperVGA standards.

character cell

A dot matrix used to produce a single character on a printer or a display screen. The matrix must be designed in such a way as to allow enough blank space between characters and lines to produce an acceptable level of legibility.

character printer

Any printer that prints one character at a time, such as a dot matrix printer or a daisy wheel.

checksum

In error checking systems, a numeric value assigned to a block of data. Checksums are capable of detecting single bit errors and some multiple bit errors.

chip

The common name for an integrated circuit (IC). Preceded by the development of the transistor, ICs can contain from several dozen to several million electronic components (resistors, diodes, transistors, and so forth) on a square of silicon approximately 1/16th to 1/2 inch wide and around 1/30th of an inch in thickness. The IC can be packaged in many different styles depending on the specific use for which it is intended. The term "chip" is synonymous with *microelectronic device*.

chip set

A group of specifically engineered ICs designed to perform a function interactively.

client/server network

Workstations (or clients) operating in conjunction with a master file server computer that controls the network.

clock

An internal timing device. Several varieties of clocks are used in computer systems. Among them are the CPU clock, the realtime clock, a timesharing clock, and a communications clock.

CMOS (Complementary Metal Oxide Semiconductor)

A MOS device fabricated by combining both P and NMOS components. It has a lower packing density than NMOS alone, but it operates at higher speed and lower power consumption than NMOS devices. CMOS ICs are very sensitive to voltage spikes and static discharges and must be protected from static shock.

CMOS diagnostics

ROM-based diagnostic routines available on the MC-2000 system board for testing hard-disk drives, floppy-disk drives, keyboards, video displays, and miscellaneous ports.

CMOS setup

A software setup program used to provide the system with information about what options are installed. The configuration information is stored in special CMOS registers that are read each time the system boots up. Battery backup prevents the information from being lost when power to the system is removed.

cold boot

Booting a computer by turning the power on.

color monitor

Also known as RGB monitors, these display types enable the user to run text and/or color-based applications such as graphics drawing and CAD programs. RGB monitors are of two basic types: digital (TTL) and analog. Analog RGB monitors enable the use of many more colors than digital RGB monitors do.

color palette

A set of VGA registers that contains digital red, green, and blue values that can be converted into analog signals to drive a color CRT's electron guns to create colors on the screen.

color printer

Any printer capable of printing in color, using thermal-transfer, dot-matrix, electrophotographic, electrostatic, ink jet, or laser printing techniques.

COM1

The label used in Microsoft DOS versions assigned to serial port #1.

compatible

A reference to any piece of computer equipment that works or looks like a more widely known standard or model. A PC compatible, or clone (such as the Marcraft Turbo-PC), is a PC that, although physically differing somewhat from the IBM-PC, runs software developed for the IBM-PC and accepts its hardware options.

composite video

The video-only portion of the standard NTSC TV signal with all video signals mixed together. Some computers offer this type of signal for use with home TV sets.

computer word

The data format that exists inside the computer. There are three types of computer words: 1) Numeric words that contain only numeric information in binary, octal, hexadecimal, or binary coded decimal (BCD) code. 2) Alphanumeric words that contain codes representing letters, numbers, and special characters. As these codes are identified, the characters are reconstructed from specific bit patterns stored in the output device. 3) Instruction words that are used to tell the computer what type of operation to perform, and where to find any data needed to complete the operation.

CONFIG.SYS file

A Microsoft DOS file that, upon startup, is used to customize the system's hardware environment. The required peripheral device drivers (with SYS file extensions) are initialized.

configuration

A customized computer system or communications network composed of a particular number and type of interrelated components. The configuration varies from system to system, requiring that some means be established to inform the system software about which options are currently installed.

continuous forms

Paper sheets that are joined together along perforated edges and used in printers that move them through the printing area with motorized sprockets. Sprockets may fit into holes on both sides of the paper.

control bus

A pathway between the MPU and the various memory, programmable, and I/O elements of the system. Control bus signals are not necessarily related to each other and can be unidirectional or bidirectional.

control character

A special type of character that causes some event to occur on a printer, display, or communications path such as a line feed, a carriage return, or an escape.

control unit

The subsection of the CPU that sends the control signals to and receives control signals from all the other parts of the system such as memory chips, programmable controllers, other bus masters, disk drives, and the various peripherals. It regulates the data traffic between all devices.

co-processor

A high-speed numeric microprocessor designed to work with a matching MPU, and usually installed in an empty IC socket specifically provided for it by the system board manufacturer. It extends the system's arithmetic and instruction set to provide for high-speed mathematical calculation capabilities.

CPU (Central Processing Unit)

The part of the computer that does the thinking. It consists of the control unit and the ALU. On personal computers, the CPU is contained on a single chip, while on a minicomputer it occupies one or several printed circuit boards. On mainframes, a CPU is contained on many printed circuit boards. Its power comes from the fact that it can execute many millions of instructions in a fraction of a second.

CRC (Cyclic Redundancy Check)

The error-checking technique that ensures communications channel integrity by utilizing division to determine a remainder. If the transmitter and receiver do not agree on what the remainder should be, an error is detected.

CRT (Cathode Ray Tube)

The vacuum tube that is used as the display screen for both TVs and computer terminals. Sometimes the term is used to mean the terminal itself.

CRT controller

The name given to the circuitry or IC responsible for providing the horizontal and vertical synchronization signals, writing screen data into the screen memory, refreshing the screen, and controlling all the display manipulation functions.

CTS (Clear To Send)

An RS-232 handshaking signal sent from the receiver to the transmitter indicating readiness to accept data.

cursor

The movable display screen symbol that indicates to the user where the action is taking place. The text cursor is usually a blinking underline or rectangle; the graphics cursor can change into any predetermined shape at different parts of the screen.

cursor keys

The keyboard cluster of keys that move the cursor around the display screen. Enhanced keyboards have two clusters of cursor keys in case the numeric keypad is in use.

cycle stealing

A mode of DMA operation whereby the DMA controller senses periods during which the bus system is not being used, and performs data transfers during these periods. In this way, both processing and peripheral operations can occur simultaneously with some degree of overlap.

cylinder

The combination of all tracks, normally on multiple-platter disk drives, that reside at the same track number location on each surface.

D

DAC (Digital to Analog Converter)

A device that converts a binary coded signal (digital) into a continuously varying signal (analog) for use by an analog device.

DACK (DMA ACKnowledge)

A signal issued by the DMA controller to the requesting I/O device indicating that a high-speed data transfer can now commence.

daisy wheel printer

A printer utilizing a print mechanism containing a plastic or metal hub with characters embossed on the ends of the spokes. The hammer waits until the correct character is spun under it, and then hits the ribbon onto the paper.

data

Information assembled in small units of raw facts and figures.

data bus

A bidirectional pathway linking the MPU to memory and I/O devices, the size of which usually corresponds to the word size of the computer.

data compression

Most compression algorithms use complex mathematical formulas to remove redundant bits, such as successive 0s or 1s from the data stream. When the modified word is played back through the decompression circuitry the formula reinserts the missing bits to return the data stream to its original state.

DCE (Data Communications Equipment)

A communications device, usually a modem, that establishes, maintains, and terminates a data transfer session. It also serves as a data converter when interfacing different transmission media.

decoder

A device similar to a demultiplexer. A binary coded input (several input lines) is used to activate a single output. The number of the activated output line corresponds with the number of the binary input code.

default

The normal action taken, or setting used, by the hardware or software when the user does not otherwise specify.

demodulator

A device that removes the data from the carrier frequency and converts it to its originally unmodulated form.

diagnostics

Software programs specifically designed to test the operational capability of the computer memory, disk drives, and other peripherals. The routines are available on disks or on ROM chips. Errors may be indicated by beep codes or visual reports. They can normally point to a board-level problem, but not down to a particular component, unless the routine has been written for a particular board being used in the system under test. A complete system failure requires a ROM-based diagnostic program as opposed to a disk-based routine.

dibit

A term used to describe a combination of two consecutive bits. A dibit can be transmitted as a single entity using phase modulation techniques.

digital

A term referring to either a device or a process that can manipulate information pulses. These pulses can take only two distinct voltage states or values: on or off, high or low. In most cases, the term "digital" is synonymous with "computer."

DIMMs

Dual in-line memory modules. DIMMs are 168-pin plug-in memory modules similar to SIMMs.

DIP (Dual In-line Package)

The most common form of IC package, available in 8-pin mini-DIPs, 14-pin, 16-pin, 20-pin, 24-pin, 28-pin, and 40-pin DIPs. Measurements taken at the various pins provide the information used by the technician when troubleshooting electronic equipment containing these digital devices.

direct access

Accessing data stored at only a specific location, without having to scan the data that occupies memory space either in front of or behind it.

direct I/O

An I/O addressing method that uses no address allocations but requires extra control lines.

directory

A collection of disk files organized under one heading and simulating a file drawer.

disk controller

The printed circuit board, normally plugged into a system board expansion slot, that contains the circuitry responsible for communicating with the disk drive.

disk drive

The peripheral storage device that reads and writes data to spinning magnetic or optical disks. The drive can either hold removable disks or contain permanent platters.

DMA (Direct Memory Access)

The capability of certain intelligent, high-speed I/O devices to perform data transfers themselves, with the help of a special IC device called a DMA controller.

DMA controller

An intelligent (programmable) IC chip specifically designed to perform high-speed data transfers when asked to do so by an I/O device. The DMA controller has the capability to take over the entire bus system for short lengths of time.

DOS (Disk Operating System)

Can be a generic term, but in most cases it refers to the Microsoft family of computer operating systems (PC-DOS for IBM equipment or MS-DOS for compatibles).

DOS prompt

A screen symbol that indicates to the user that the system is ready for a command. It usually consists of the current drive letter, followed by a colon and a blinking cursor.

dot pitch

A measurement of the resolution of a dot matrix. The width of an individual dot in millimeters describes a display's resolution, with the smaller number representing the higher resolution. The number of dots per linear inch describes a printer's resolution, with the higher number representing the higher resolution.

dot-matrix printer

A type of printer that forms its images out of one or more columns of dot hammers. Higher resolutions require a greater number of dot hammers to be used.

DPSK (Differential Phase Shift Keying)

A common form of phase modulation in modems in which the phase is incrementally altered to reflect multiple bit combinations, yielding multiple bits per baud of bandwidth. See PSK.

DRAM (Dynamic Random Access Memory)

A type of RAM that loses its data, regardless of power considerations, unless it is refreshed at least once every 2 milliseconds.

DREQ (DMA REQuest)

A peripheral signal sent to the DMA controller requesting a high-speed data transfer without MPU intervention.

DSR (Data Set Ready)

An RS-232 handshaking signal sent from the modem to its own computer indicating its readiness to accept data.

DTE (Data Terminal Equipment)

A piece of data communications equipment that is serving as either the source or destination of a data transfer. A DTE is normally a computer terminal.

DTR (Data Terminal Ready)

An RS-232 handshaking signal that is sent to a modem by its own computer to indicate a readiness to accept data.

E

edge connector

The often double-sided row of etched lines on the edge of an adapter card that plugs into one of the computer's expansion slots.

EEPROM (Electrically Erasable Programmable Read-Only Memory)

A type of nonvolatile semiconductor memory device that enables erasure and reprogramming from within a computer using special circuitry. These devices enable specific memory cells to be manipulated, rather than requiring a complete reprogramming procedure as in the case of EPROMs.

EGA (Enhanced Graphics Adapter)

The IBM video display standard providing medium-resolution text and graphics and requiring a digital RGB Enhanced Color Display monitor. This standard has been superseded by VGA.

EIA (Electronics Industries Association)

An organization, founded in 1924, made up of electronic parts and systems manufacturers. It sets electrical and electronic interface standards such as the RS-232C.

EISA (Extended Industry Standard Architecture)

A PC bus standard that extends the AT bus architecture to 32 bits and enables older PC and AT boards to plug into its slot. It was announced in 1988 as an alternative to the IBM Micro Channel.

electron beam

A continuous stream of electrons specifically directed toward a receiving target.

electron gun

The device by which the fine beam of electrons is created that sweeps across the phosphor screen in a CRT.

EMI (ElectroMagnetic Interference)

A system-disrupting electronic radiation created by some other electronic device. The FCC sets allowable limits for EMI, in Part 5 of its Rules and Regulations. Part A systems are designed for office and plant environments, and Part B systems are designed for home use.

EMM (Expanded Memory Manager)

Any software driver that permits and manages the use of expanded memory in 386 and higher machines.

EMS (Expanded Memory Specification)

A method of using memory above one megabyte on computers using DOS. Co-developed by Lotus, Intel, and Microsoft, each upgrade has allowed for more memory to be used. EMS is dictated by the specific application using it. In 286 machines, EMS is installed on an adapter card and managed by an EMS driver. See *EMM.*

EOP (End OPeration)

A control line activated by an I/O device to indicate the termination of a DMA transfer.

EPROM (Erasable Programmable Read-Only Memory)

A type of nonvolatile semiconductor memory device that can be programmed more than once. Selected cells are charged using a comparatively high voltage. EPROMs can be erased by exposure to a source of strong ultraviolet light, at which point they must be completely reprogrammed.

ergonomics

The study of people-to-machine relationships. A device is considered to be ergonomic when it blends smoothly with a person's body actions.

error checking

The act of testing the data transfer in a computer system or network for accuracy.

esc key (escape key)

This keyboard key is used to cancel an application operation or to exit some routine.

escape character

Used in conjunction with other codes to perform specific actions. It is a control code with a numeric value of 27 in ASCII.

ESDI (Enhanced Small Device Interface)

A hardware interface standard for connecting disk and tape drives and computers, enabling these devices to hold up to 1GB of data with a transfer rate between one to three megabytes per second.

execution cycle

The second part of a two-step process that occurs repeatedly until a program is completely executed by the computer. During an execution cycle the instruction, which was fetched, decoded, and identified during the instruction cycle, is carried out. Another instruction cycle is entered immediately after an execution cycle, unless the current execution cycle is the last.

execution time

The time required for a computer to execute a single instruction, and making up the last half of an instruction cycle.

expansion slot

The receptacle mounted on the system board into which adapter cards are plugged to achieve system expansion. The receptacle interfaces with the I/O channel and system bus; therefore the number of slots available determines the expansion potential of the system.

extended memory

The memory above one megabyte in Intel 286 and higher computers, and used for RAM disks, disk caching routines, and for locating the operating system files in recent versions of Microsoft DOS.

external command

A DOS command that requires the loading of its COM or EXE file prior to its execution.

F

FAT (File Allocation Table)

The part of the DOS file system that keeps track of where specific data is stored on the disk.

FDC (Floppy Disk Controller)

A specialized IC that provides an intelligent interface between the system unit and the disk drive unit.

FDISK command

The DOS utility program that permits the partitioning of the hard disk into several independent disks.

FDM (Frequency Division Multiplexing)

A widely-used data transmission method used to send multiple signals over a single channel. Each signal is modulated onto a separate carrier.

FIFO (First In First Out)

A data storage method that retrieves the data that has been stored for the longest time.

file

Any program, record, table, or document that is stored under its own filename.

filename

A name created by the user or programmer to identify a file. It must adhere to some specific operating system rules for naming a file.

firmware

A term used to describe the situation in which programs (software) are stored in ROM ICs (hardware) on a permanent basis.

floppy disk

A removable secondary storage medium for computers, composed of flexible magnetic material and contained in a square envelope or cartridge. A floppy disk can be recorded and erased hundreds of times.

FM (Frequency Modulation)

1) A method of data transmission that modulates the data into a carrier by varying the carrier frequency.

2) A method of recording data onto magnetic medium using one clock bit with each bit of data recorded.

font

One set of alphanumeric characters possessing matching design characteristics such as typeface, orientation, spacing, pitch, point size, style, and stroke weight.

form feed

The moving of the next paper form into the proper print position, accomplished either by pressing the form feed (FF) button on the printer or by sending the printer the ASCII form feed character.

FORMAT command

A Microsoft DOS utility that prepares a disk for use by the system. Track and sector information is placed on the disk, and bad areas are marked so that no data will be recorded on them.

formatting

The act of preparing a hard or floppy disk by using the FORMAT command.

frame

One screen of computer graphics data, or the amount of memory required to store it.

FRU (Field Replaceable Unit)

The portions of the system that can be conveniently replaced in the field.

FSK (Frequency Shift Keying)

A simple, data transmission modulating technique that uses only two frequencies, one for a logic 1, and the other for a logic 0.

full-duplex

A method of data transmission that enables data to flow in both directions simultaneously.

function keys

A special set of keyboard keys used to give the computer special commands. They are frequently used in combination with other keys, and can have different uses depending on the software application being run.

G

gap

1) The space between the blocks of data on disk or tape.

2) The space in the R/W head over which magnetic flux passes when data transfer occurs.

GHz (GigaHertz)

One billion hertz or cycles per second.

graphics

The creation and management of pictures using a computer.

graphics controller

The portion of the VGA video controller that supplies an interface between the video memory and the attribute controller during active display periods. It also acts as an interface between the system and the video memory when the display is being updated. See *attribute controller*.

ground

1) Any point from which electrical measurements are referenced.

2) Earth ground is considered to be an electrical reference point of absolute zero, and is used as the electrical return path for modern power transmission systems. This ground, often incorporated by electronic devices to guard against fatal shock, is called chassis or protective ground.

3) An actual conductor in an electronic circuit being used as a return path, alternately called a signal ground.

GUI (Graphical User Interface)

A form of operating environment that uses a graphical display to represent procedures and programs that can be executed by the computer.

H

handset

The speaker and microphone portion of the telephone.

handshaking

A system of signal exchanges conducted between the computer system and a peripheral device during the data transfer process. The purpose of these signals is to produce as orderly a flow of data as possible.

hard disk

A metal disk for external storage purposes, coated with ferromagnetic coating and available in both fixed and removable format.

hardware

Any aspect of the computer operation that can be physically touched. This includes IC chips, circuit boards, cables, connectors, and peripherals.

hexadecimal (hex)

Meaning sixteen; a numbering system with a base of sixteen. It provides a quick shorthand method of representing all the possible values of a byte. Each nibble (4 bits) represents one hex digit. The hex digits themselves are composed consecutively of arabic numerals 0 through 9, followed by uppercase letters A through F.

HGC (Hercules Graphics Card)

The monochrome graphics standard developed by Hercules Computer Technology to meet the needs of early PC users for graphics capabilities. Introduced in 1982, it provides a text and graphics resolution of 720 × 348 pixels.

high-level language

A programming language that is machine-independent, such as BASIC, COBOL, FORTRAN, Pascal, and C. When using a high-level language, the programmer can concentrate on the logic of the problem to be solved without having to be concerned about the specific architecture of the machine.

high-resolution

A term used to describe a printer or display screen image of high quality. Actual quality depends on the number of dots used per square inch to create the image.

I

IC (Integrated Circuit)

The technical name for a chip. See *chip*.

IDE (Integrated Drive Electronics)

A method of disk drive manufacturing that locates all the required controller circuitry on the drive itself, rather than on a separate adapter card. Also known as AT Attachment interface.

impact printer

Any printer that produces a character image by hammering onto a combination of embossed character, ribbon, and paper.

ink-jet printer

A high-resolution printer that produces its image by spraying a specially treated ink onto the paper.

input device

Any computer input-generating peripheral device such as a keyboard, mouse, light pen, scanner, or digitizer.

instruction cycle

The first part of a two-step process that occurs repeatedly until a program is completely executed by the computer. During an instruction cycle an opcode is fetched, decoded, and identified.

instruction word

A class of binary coded data word that tells the computer what operation to perform and where to find any data needed to perform the operation.

INT (INTerrupt)

A signal sent to the MPU from an I/O device, occurring randomly, which is capable of interrupting the MPU during program execution. An interrupt is usually generated when an input or output is required.

INTA (INTerrupt Acknowledge)

A signal sent on the control bus by the MPU to indicate that it has received an interrupt request.

intelligent controller

Usually an IC, or series of ICs, with built-in microprocessor capabilities dedicated to the controlling of some peripheral unit or process. Single-chip controllers are sometimes referred to as smart chips.

interface

The joining of dissimilar devices so that they function in a compatible and complementary manner.

interlaced

The method of rewriting the monitor screen repeatedly, and by alternately scanning every other line and then the unscanned every other line.

Internet

The most famous wide area network is actually a network of networks operating together. The main communication path is a series of networks established by the U.S. government. This backbone has expanded around the world and offers access to computers in every part of the globe.

interrupt controller

A special programmable IC responsible for coordinating and prioritizing interrupt requests from I/O devices, and sending the MPU the starting addresses of the interrupt service routines so that the MPU can service the interrupting device and then continue executing the active program.

interrupt vector

One of a possible 256 4-byte address pointers that reside in the first kilobyte of memory in a PC-based system. Each vector points to the location, in another area of memory, where a specific interrupt service routine is stored. These routines contain instructions for handling a wide variety of peripheral activities and internal system functions.

I/O (Input/Output)

A type of data transfer occurring between a CPU and a peripheral device. Whenever any data transfer occurs, output from one device becomes an input to another.

I/O channel

The actual high-speed pathway between the peripheral device and the computer.

I/O interface

The connecting channel between the CPU and a I/O device in a large computer. In a small computer, the interface also includes the interface controller and the connecting cable between the computer and the I/O device.

I/O port

The external window or connector on a computer, used to effect an interface with a peripheral device. The I/O port may appear as either parallel data connections or serial data connections.

IRQ (Interrupt ReQuest)

See *INT*.

ISA (Industry Standard Architecture)

A term that refers to the bus structures used in the IBM PC series of personal computers. The PC and XT use an 8-bit bus, and the AT uses a 16-bit bus.

ISPs (Internet Service Providers)

Companies that provide the technical gateway to the Internet. They connect all the users and individual networks together.

J

joystick

A computer input device that offers quick, multi-directional movement of the cursor for CAD systems and video games.

jumper

Normally, a 2- or 4-pin BERG connector, located on the system board or an adapter card, that permits the attachment of a wired hardware switch or the placement of a shorting bar to affect a particular hardware function or setting.

K

keyboard

The most familiar computer I/O device, incorporating a standard typewriter layout with the addition of other specialized control and function keys.

keyboard controller

Usually a dedicated MPU, located in the keyboard itself, that translates a key closure into the appropriate scan code.

L

laser printer

Any printer that utilizes the electrophotographic method of image transfer. Light dots are transferred to a photosensitive rotating drum that picks up electrostatically charged toner before transferring it to the paper.

LCD (Liquid Crystal Display)

The type of output display created by placing liquid crystal material between two sheets of glass. A set of electrodes is attached to each sheet of glass. Horizontal (row) electrodes are attached to one glass plate, while vertical (column) electrodes are fitted to the other plate. These electrodes are transparent and let light pass through. A pixel is created in the liquid crystal material at each spot where a row and a column electrode intersect. When the pixel is energized, the liquid crystal material bends and prevents light from passing through the display.

LED (Light Emitting Diode)

A particular type of diode that emits light when conducting, and used in computers and disk drives as active circuit indicators.

letter quality

Refers to a print quality as good or better than that provided by an electric typewriter.

light pen

An I/O device meant to be used in conjunction with a video terminal in the selection of items from a screen menu or the drawing of images directly. The pen, when activated by the user, senses the scanning CRT electron beam during screen refresh, and thereby is able to tell the system its onscreen location.

line feed

A character control code (LF) used to kick the printer down to the next line or to move the screen cursor down to the next line. Some printers have a line feed button that accomplishes the same action.

line printer

An expensive and high-speed printer, characterized by the capability to print a line of text all at once.

loopback

A modem test procedure that enables a transmitted signal to be returned to its source for comparison with the original data.

low-level format

The procedure that initializes a hard-disk drive according to a sector layout plan dictated by its controller.

low resolution

A term used to describe a printer or display screen image of low quality, due to the limited number of dots used per square inch to create the image.

LPT1

The label used in Microsoft DOS versions assigned to parallel port #1, usually reserved for printer operation.

LSB (Least Significant Bit)

The bit occupying the rightmost position in the byte or word.

LSI (Large Scale Integration)

IC devices containing a large number of electronic components (from 3,000 to 100,000 approximately).

luminance

A screen pixel's degree of brightness, measured in lumens.

M

machine instruction

Any instruction that requires no further translation in order for the computer to carry it out. It is composed of an op code and any number of operands.

magnetic disk

The most popular form of secondary data storage for computers. Shaped like a platter, and coated with an electromagnetic material, magnetic disks provide direct access to large amounts of stored data, and can be erased and rerecorded many times.

magnetic tape

The most popular form of secondary data storage backup for computers. Because access to data is sequential in nature, magnetic tape is primarily used to restore a system that has suffered a catastrophic loss of data from its hard-disk drive.

maskable interrupts

A specific class of interrupts that the computer can ignore under certain conditions.

MCA (Micro Channel Architecture)

The 32-bit bus introduced by IBM on the high end of its PS/2 line of computers. MCA is totally incompatible with the original PC bus.

MDA (Monochrome Display Adapter)

The original video display standard introduced in the IBM PC line, having only monochrome text capabilities (no graphics).

memory management

Methodology used in handling a computer's memory resources, including bank switching, memory protection, and virtual memory.

memory map

A layout of the memory and I/O device addressing scheme used by a particular computer system.

memory-mapped I/O

An I/O addressing method where I/O devices are granted a portion of the available address allocations, thus requiring no additional control lines to implement.

memory unit

The section of the digital computer where instructions and data to be used by the computer are stored. Volatile memory that interfaces directly with the MPU is called internal memory; nonvolatile memory stored on peripheral devices is called external memory.

menu

A screen display of available program options or commands that can be selected through keyboard or mouse action.

MFM (Modified Frequency Modulation)

An improvement over early FM magnetic disk recording techniques that reduces the number of required synchronization bits.

MGA (Monochrome Graphics Adapter)

The Hercules-compatible adapter card containing output connectors for both a monochrome monitor and a parallel printer.

MHz (MegaHertz)

One million hertz, or cycles per second.

microcomputer

Meaning the same thing as a personal computer, or a computer using an MPU as its CPU.

MIO card (Multi-Input/Output card)

The I/O adapter card that contains a floppy disk controller capable of running two floppy drives, two serial ports, one parallel printer port, one game port, and a clock/calendar circuit.

modem (MOdulator-DEModulator)

Also called a DCE device, it is used to interface a computer or terminal to the telephone system for the purpose of conducting data communications between computers often located at great distances from each other.

monitor

1) Another name for a computer display.

2) Any hardware device or software program that checks, reports about, or automatically oversees a running program or system.

monochrome

Used to describe a computer display capable of operating with only one foreground and one background color.

MOS (Metal Oxide Semiconductor)

A category of logic and memory chip design that derives its name from the use of metal, oxide, and semiconductor layers. Among the various families of MOS devices are PMOS, NMOS, and CMOS. The first letter of each family denotes the type of construction used to fabricate the chip's circuits. MOS families do not require a highly regulated +5VDC power supply, as TTL devices do.

mouse

A popular computer I/O device used to point or draw on the video monitor. As the mouse is rolled along a desktop, the cursor moves on the screen in a corresponding manner.

MSB (Most Significant Bit)

The bit occupying the leftmost position in the byte or word.

MSI (Medium-Scale Integration)

IC devices containing a small number of electronic components (from 100 to 3,000 approximately). Such ICs include the digital logic circuits as multiplexers, demultiplexers, and decoders.

MTBF (Mean Time Between Failures)

Corresponds to the average working time for a component before failure occurs. It is equal to the total number of hours tested divided by the total number of failures.

multimedia

A term applied to a range of applications that bring together text, graphics, video, audio, and animation to provide interactivity between the computer and its human operator.

multitasking

The capability of a computer system to run two or more programs simultaneously.

MUX (Multiplexer)

A device that accepts many different digital logic input signals and passes only one at a time to its single output line. It is also known as a data selector.

N

NAK (Negative AcKnowledge)

A data communications code used by a receiver to tell the transmitter that the last message was not properly received.

nibble

A 4-bit binary pattern, which can easily be converted into a single hexadecimal digit.

NLQ (Near Letter Quality)

A quality of printing nearly as good as an electric typewriter. The very best dot-matrix printers can produce NLQ.

NMI (Non-Maskable Interrupt)

A type of interrupt that cannot be ignored by the MPU during program execution. Three things can cause a non-maskable interrupt to occur: 1) a numeric coprocessor installation error 2) a RAM parity check error 3) an I/O channel check error.

non-impact printer

Any printer that does not form its characters by using a hammer device to impact the paper, ribbon, or embossed character.

non-volatile memory

Memory, such as ROM, that is not lost after the power is turned off.

NOP (No OPeration)

A machine code instruction used to take up one byte of memory when contained in a program, or to waste some execution time during certain operations such as HALT DMA mode data transfers. Although the MPU is idling, it may still refresh its memory to prevent a system crash during the DMA operation.

NTSC (National Television Standards Committee)

This organization created the television standards in the United States, and is administered by the FCC.

NuBus

A type of expansion slot bus. NuBus is a Texas Instruments specification that is widely used in Apple Macs. It uses intelligent cards that communicate with the system during startup to determine what type of card is in the slot.

null modem cable

A cable meeting the RS-232C specification, which cross-connects two computers through their serial ports by transposing the transmit and receive lines. They must be physically located close to one another, thereby eliminating the need for a modem.

O

octal

A method of representing binary numbers by grouping them into 3-bit patterns. Because this system uses a base of 8, the highest digit that can be represented is a 7, or 111 in binary.

odd parity

The form of parity checking where the parity bit is used in order to make the total number of 1s contained in the character an odd number.

off-hook

A condition existing on a telephone line that is now capable of initiating an outgoing call, but unable to receive an incoming call.

offline

Any computer system or peripheral device that is not ready to operate, not connected, not turned on, or not properly configured.

on-hook

A condition that exists on any telephone line that is capable of receiving an incoming call.

online

Any computer system or peripheral device that is not only powered up, but also ready to operate.

op code (OPeration Code)

That part of an instruction word telling the computer what to do. The operation code is usually a verb, such as input, add, or branch.

operand

That data upon which the given operation is to be carried out. Can also refer to a specific peripheral device for I/O operations.

operand address

That data containing the memory address where the operand is located.

operating system

A special software program, first loaded into a computer at power up, and responsible for running it. The operating system also serves as the interface between the machine and other software applications.

optical mouse

A mouse that emits an infrared light stream to detect motion as it is moved around a special x-y matrix pad.

output device

Any peripheral device, such as a monitor, modem, or printer, that accepts computer output.

P

page printer

Any printer capable of printing a page at a time, such as a laser printer.

parallel interface

The multi-line channel through which the simultaneous transfer of one or more bytes occurs.

parallel mode

The mode of data transfer where an entire word is transferred at once, from one location to another, by a set of parallel conductors.

parallel port

The external connector on a computer that is used to effect an interface between the computer and a parallel peripheral such as a printer.

parity bit

Used for error checking during the sending and receiving of data within a system and from one system to another. The parity bit's value depends on how many 1 bits are contained in the byte it accompanies. See *even parity* and *odd parity*.

parity checking

A method to check for data transmission errors by using a ninth bit to ensure that each character sent has an even (even parity) or odd (odd parity) number of logic 1s before transfer. The parity bit is checked for each byte sent.

parity error

This error occurs when a data transfer cannot be verified for integrity. At least one data bit or the parity bit has toggled during the transfer process.

PC bus

Refers to the bus architectures used in the first IBM PCs, the original 8-bit bus, and the 16-bit bus extension used with the AT.

PCI (Peripheral Component Interconnect) bus

A low-cost, high-performance 32-/64-bit local bus developed jointly by IBM, Intel, DEC, NCR, and Compaq.

PDS (Processor Direct Slot) bus

Expansion slots that grant peripheral devices direct access to the microprocessor buses. The various PDS slots consist of between 90 and 140 pins. The PDS bus goes directly into the microprocessor's address, data, and control buses without any buffering. It is a fast expansion slot. It is not standardized between different MAC models.

peer-to-peer network

A network of computers that can act individually and share information and resources with the other units on the network.

pel (Picture ELement)

See *pixel*.

peripherals

Also called I/O devices, these units include secondary memory devices such as hard-disk drives, floppy-disk drives, magnetic-tape drives, modems, monitors, mice, joysticks, light pens, scanners, and even speakers.

peripheral controller

A dedicated IC or circuit that interprets system commands to its unit and sends it the required control signals.

persistence

The amount of time that a CRT phosphor dot continues to emit light after being struck by an electron beam.

pin feed

A method of moving continuous forms through the print area of a printer by mounting pins on each side of a motorized platen to engage the holes on the right and left side of the paper.

PISO (Parallel In Serial Out)

Refers to a type of shift register that loads a parallel word in a single clock pulse. After the word has been loaded into the register, the bits are shifted out, one clock pulse per bit.

pitch

A unit of measurement for print type, it lists the number of characters per inch.

pixel

Also called a pel, or picture element, it is the smallest unit (one dot for monochrome) into which a display image can be divided.

pointing device

Any input device used for the specific purpose of moving the screen cursor or drawing an image.

polarizer

An optical device that either blocks or allows the passage of light through it depending upon the polarity of an electrical charge applied to it.

polling

A system of initiating data transfer between a computer system and a peripheral, where the status of all the peripherals is examined periodically under software program control by having the MPU check the READY line. When it is activated by one of the peripherals, the MPU begins the data transfer by using the corresponding I/O port.

POST (Power On Self Tests)

A group of ROM BIOS diagnostic tests that are performed on the system each time it is powered up.

power supply

The component in the system that converts the AC voltage from the wall outlet to the DC voltages required by the computer circuitry.

preventive maintenance

Any regularly scheduled checking and testing of hardware and software with the goal of avoiding future failure or breakdown.

printer

A peripheral device for the printing of computer text or graphics output.

printer buffer

A special-purpose memory area or device that accepts high-speed transfer of print data from the computer, and transfers the data to the printer at printer-compatible speeds, freeing the computer for other tasks during the printing process.

printer font

A prescribed character set, properly formatted for use by the printer.

program

Any group of instructions designed to command a computer system through the performance of a specific task. Also called software.

programmed I/O

A system of initiating data transfer between a computer system and a peripheral, where the MPU alerts the specific device by using an address call. The I/O device can signal its readiness to accept the data transfer by using its BUSY line. If BUSY is active, the MPU can perform other tasks until the BUSY line is deactivated, as which time the transfer can begin.

programming language

Any one of a number of languages used to write computer instructions in symbolic form, without regard to a machine's specific hardware makeup.

prompt

A software-supplied message to the user, requiring some specific action or providing some important information. It can also be a simple symbol, indicating that the program is successfully loaded and waiting for a command from the user.

protected mode

An operational state that enables a 286 or higher computer to address all its memory.

protocol

A set of rules that govern the transmitting and receiving of data communications.

PSK (Phase Shift Keying)

A method of modulation in modems through which the phase of the signal is altered to reflect the transmission of 0s and 1s. See *DPSK*.

Q

QAM (Quadrature Amplitude Modulation)

A data communication transmission method, combining both AM and PSK modulation techniques, that provides for much higher bit transfer rates than either system does alone.

queue

A special and temporary storage (RAM or registers) area for data in printing or internal program execution operations.

QWERTY keyboard

A keyboard layout that was originally designed to prevent typists from jamming old-style mechanical typewriters, it is still the standard English language keyboard. The name spells out the first six leftmost letters in the first alphabetic row of keys.

R

RAM (Random Access Memory)

A type of semiconductor memory device that holds data on a temporary or volatile basis. Any address location in the RAM memory section can be accessed as quickly as any other location.

RAMDAC

A section of the VGA video controller circuitry that contains the RAM color palette and three DACs. The DACs translate the coded digital values stored in the color palette into corresponding analog output levels. These voltage levels drive the CRT's electron guns to produce a corresponding color using a mixture of red, green, and blue. When the red, green, and blue signals are combined, each color produced on the screen relates to an 18-bit value. This produces a maximum of 262,144 (2^{18}=262,144) or 256K colors.

raster graphics

A graphics representation method that uses a dot matrix to compose the image.

raster scan

The display of a video image, line by line, by an electron beam deflection system.

read

Any act of inputting prerecorded digital data, whether it be on disk, tape, or in ROM or RAM.

read-only

1) A file parameter setting that prevents a file from being altered.

2) Refers to data that is permanently stored on the media or to such media itself.

ready

A control bus signal line, used by an I/O device, to cause the microprocessor to suspend its internal operations, temporarily, until it is prepared to engage in data transfer.

read/write head

Usually abbreviated "R/W head," the device by which a disk or tape drive senses and records digital data on the magnetic medium.

real mode

A mode of operation in 286 and higher machines in which the computer functions under the same command and addressing restrictions as an 8086 or 8088.

reboot

To restart the computer or to reload the operating system.

refresh

A required method of reenergizing a memory cell or display pixel for its data to be held continually.

RESET

A control bus signal, activated either by a soft or hard switch, which sets the system MPU and all programmable system devices to their startup, or initialization, values. This enables the computer to begin operation following the application of the RESET input signal.

resolution

A measurement of the sharpness of an image or character, either of a printer or a display monitor. For a monitor, resolution consists of the number of dots per scan line, times the number of scans per picture. For a printer, resolution consists of the number of dots present per linear inch of print space.

reverse video

A method commonly used to highlight specific screen characters by reversing their foreground and background colors.

RGB monitor

Any video display, either analog or digital, requiring separate red, green, and blue input signals from the computer.

RLL (Run Length Limited)

An improvement over the MFM magnetic disk encoding technique, enabling more data to be packed into the same space.

ROM (Read Only Memory)

A type of semiconductor memory device that holds data on a permanent or nonvolatile basis.

ROM BIOS

A collection of special programs (native intelligence) permanently stored in one or two ROM ICs installed on the system board. These programs are available to the system as soon as it is powered up, providing for initialization of smart chips, POST tests, and data transfer control.

RS-232C

The most widely used serial interface standard, it calls for a 25-pin D-type connector. Specific pins are designated for data transmission and receiving, as well as a number of handshaking and control lines. Logic voltage levels are also established for the data and the control signals on the pins of the connector.

RS-422

An enhancement to the original RS-232C interface standard and adopted by the EIA, it uses twisted-pair transmission lines and differential line voltage signals resulting in higher immunity for the transmitted data.

RS-423

Another enhancement to the original RS-232C interface standard and adopted by the EIA, it uses coaxial cable to provide extended transmission distances and higher data transfer rates.

RTS (Request To Send)

The RS-232 handshaking signal sent from the originating modem to the receiving modem requesting permission to begin the data transmission.

S

scan code

A code supplied by the keyboard MPU that corresponds to a keypress from the keyboard.

scan rate

The total number of times per second that a video raster is horizontally scanned by the CRT's electron beam.

SCSI (Small Computer System Interface)

An interface standard adapter card for PCs that can provide high-speed data transfer control (4MB/sec) for up to seven devices, while occupying only one expansion slot. Apple was the first personal computer maker to select the SCSI interface as the bus standard for peripheral equipment. The MAC's SCSI port can be daisy-chained to enable it to connect up to six external peripherals to the system. Even though a total of eight SCSI device address numbers are possible, only six are available for external devices.

sector

One of many individual data-holding areas into which each track of a disk is divided during the format process.

sector interleave

A method by which a disk drive's read and write operations can be optimized. The interleave number indicates the order in which the sectors are to contain the data pertaining to the same file.

seek time

The amount of time required for a drive's R/W head to settle over a particular disk track following an I/O command.

sequential access

A data access method in which all data existing in the media ahead of the desired data must be sequentially checked and identified.

serial interface

A channel through which serial digital data transfer occurs. Although multiple lines may be used, only one actually carries the data. The most popular serial interface standard is the EIA RS-232C.

serial mode

The mode of data transfer in which the word bits are transferred one bit at a time, along a single conductor.

serial mouse

A type of mouse that plugs into a serial port rather than an adapter card.

serial port

The external connector on a computer that is used to effect an interface between the computer and a serial device such as a modem. A typical serial port uses a DB-25 or a DB-9 connector.

serial transmission

The transmission of data one bit at a time, one bit following the next.

shadow RAM

An area of RAM used for copying the system's BIOS routines from ROM. Making BIOS calls from the RAM area improves the operating speed of the system. Video ROM routines are often stored in shadow RAM also.

SIMM (Single Inline Memory Module)

A memory chip, circuit board module, containing eight (without parity) or nine (with parity) memory chips, and designed to plug into special sockets.

simplex

A term used to describe one-way data transmission.

SIPO (Serial In Parallel Out)

Refers to a type of shift register that loads a serial word in, one bit per clock pulse. After the bits have been loaded into the register they are immediately available to the system, and may require tri-state buffering to isolate them from the data bus.

soft-sectored

A type of storage disk noted for its programmed method of sector identification.

software

Any aspect of the computer operation that cannot be physically touched. This includes bits, bytes, words, and programs.

software interrupt

Any interrupt caused by a program instruction.

speaker

The computer system's audio output device. Measuring 2 1/4 inches in diameter, and rated at 8 ohms, 1/2 watts, the speaker is usually used as a system prompt and as an error indicator. It is also capable of producing arcade sounds, speech, and music.

spindle

The center spinning shaft in a disk drive unit that fits through and grasps the disks during drive operation.

SRAM (Static Random Access Memory)

A type of RAM that can store its data indefinitely as long as power to it is not interrupted.

SSI (Small Scale Integration)

IC devices containing a very small number of electronic components (from 2 to 100 approximately). Such ICs include the basic digital logic gates such as AND, OR, NAND, NOR, and INVERTER circuits.

ST506

For hard-disk drives up to 40MB in size, this hard-disk controller for PCs uses MFM encoding techniques with data transfer rates of 500Kbps.

ST506 RLL (ST506 Run-Length Limited)

An improved version of the ST 506 hard-disk controller for PCs, enabling the use of higher-capacity disks and transfer rates of 750Kbps.

stack

A special memory storage area that is used to keep track of internal program operations during interrupt requests, or when the program being executed contains calls or subroutines that jump to nonsequential memory locations. The address information, needed to return to the proper point in the program after the subroutine, is contained in the stack. The stack operates on a LIFO (Last In First Out) basis; the last address pushed onto the stack is the first address popped off.

stack pointer

A special memory address register used in the operation of the stack memory area to record the location of the next push or pull operation. See *stack*.

start bit

In asynchronous serial data transmission, this bit denotes the beginning of a character and is always a logic low pulse, or space.

static electricity

A stationary charge of electricity normally caused by friction, and potentially damaging to sensitive electronic components. It can be a serious problem in environments of low humidity.

stepper motor

A special class of motor used in disk drive access arms and dot matrix print head mechanisms. Very precise movements are obtained by applying voltage pulses, one pulse per step, to the motor control circuitry.

stop bit

The bit, sent after each character in an asynchronous data communications transmission, that signals the end of a character.

sync character

A character that is transmitted in synchronous data communications to synchronize the timing between the transmitter and receiver.

synchronous transmission

A method of serial data transmission in which both the transmitter and the receiver are synchronized by a common clock signal.

system board

The large printed circuit board (mother board) into which peripheral adapter boards (daughter boards) may plug, depending on the number of devices working with the system. The system board is populated with 100 or more IC chips, depending on how much on-board memory is installed. As well as RAM chips, the system board contains the MPU, BIOS ROM, several programmable controllers, system clock circuitry, switches, and various jumpers. Also, most system boards come with an empty socket into which the user may plug a compatible co-processor chip to give the computer some high-level number crunching capabilities.

system-level interface

An interface that enables the system to directly access the I/O device through an expansion slot without an intermediate interface circuit. The system is isolated from the peripheral device and sees only its logical configuration.

system software

A class of software dedicated to the smooth control and operation of a computer system and its various peripherals.

system unit

The main computer cabinet containing the primary components of the system. This includes the main logic board (system or mother board), disk drive(s), switching power supply, and the interconnecting wires and cables.

T

tape drive

The unit that actually reads, writes, and holds the tape being used for backup purposes.

task switching

The changing of one program or application to another either manually by the user, or under the direction of a multitasking operating system environment.

TDM (Time Division Multiplexing)

A method of sending many individual digital data signals over one high-speed data channel by sharing the amount of time allotted to each signal sequentially.

text mode

Refers to a screen display mode that displays only text (no graphics), or to any application mode enabling the entry and editing of text.

thermal printer

A printer requiring the use of specially treated paper and electrically heated print pins, which are selectively pushed against the paper to produce areas of darkness at the contact points. A thermal printer is considered a non-impact type and is very quiet, producing a print image that is medium to low resolution in quality.

toner

A form of powdered ink that accepts an electrical charge in laser printers and photocopying machines. It adheres to a rotating drum containing an image that has been oppositely charged. The image is transferred to the paper during the printing process.

track

A single disk or tape data storage channel, upon which the R/W head places the digital data in a series of flux reversals. On disks, the track is a concentric data circle; on tapes, it is a parallel data line.

trackball

1) A pointing device that enables the user to control the position of the cursor on the video display screen by rotating a sphere (trackball).

2) The sphere inside certain types of mice on which the mouse rides. As the mouse moves across a surface, the trackball rolls, creating X-Y movement data.

tractor feed

A paper-feeding mechanism for printers that use continuous forms. The left and right edges of the forms contain holes through which the tractor pins pull the paper through the print area.

transceiver

A device capable of both transmitting and receiving analog or digital signals. In computers, an example of a transceiver is the 74LS245 digital octal transceiver IC.

transmit

Although this term usually means to send data between a transmitter and receiver over a specific communications line, it can also describe the transfer of data within the internal buses of a computer or between the computer and its peripheral devices.

U

UART (Universal Asynchronous Receiver Transmitter)

A serial interface IC used to provide for the parallel-to-serial and serial-to-parallel conversions required for asynchronous serial data transmission. It also handles the parallel interface to the computer's bus, as well as the control functions associated with the transmission.

URL (Uniform Resource Locator)

A unique address on the World Wide Web used to access a web site.

USART (Universal Synchronous Asynchronous Receiver Transmitter)

A serial interface IC used to provide for the parallel-to-serial and serial-to-parallel conversions required for both asynchronous and synchronous serial data transmission. It also handles the parallel interface to the computer's bus, as well as the control functions associated with the transmission.

USRT (Universal Synchronous Receiver Transmitter)

A serial interface IC used to provide for the parallel-to-serial and serial-to-parallel conversions required for synchronous serial data transmission only.

utility program

A term used to describe a program designed to help the user in the operation of the computer.

V

vertical refresh

A rate or frequency, in times per second, that the entire display screen is rewritten.

VESA (Video Electronics Standards Association) bus

A 64-bit local bus standard developed to provide a local bus connection to a video adapter. Its operation has been defined for use by other adapter types, such as drive controllers, network interfaces, and other hardware.

VGA (Video Graphics Array)

Another video standard, developed by IBM, providing medium and high text and graphics resolution. It was originally designed for IBM's high-end PS/2 line, but other vendors have created matching boards for PC and AT machines also, making it the preferred standard at this time. Requiring an analog monitor, it originally provided 16 colors at 640×480 resolution. Third-party vendors have boosted that capability to 256 colors, while adding an even greater 800×600 resolution, calling it Super VGA.

video adapter

Sometimes referred to as a display adapter, graphics adapter, or graphics card, it is a plug-in peripheral unit for computers, fitting in one of the system board option slots, and providing the interface between the computer and the display. The adapter usually must match the type of display (digital or analog) with which it is used.

video capture

Software used to capture frames of television video and convert it into digital formats that can be processed by the computer.

virtual disk

A method of using RAM as if it were a disk.

virtual memory

A memory technique that enables several programs to run simultaneously, even though the system does not have enough actual memory installed to do this. The extra memory is simulated.

VLSI (Very Large Scale Integration)

IC devices containing a very large number of electronic components (from approximately 100,000 to 1,000,000).

volatile memory

Memory (RAM) that loses its contents as soon as power is discontinued.

VOM (Volt Ohm Milliammeter)

A basic piece of electronic troubleshooting equipment that provides for circuit measurements of voltage, current, and resistance in logarithmic analog readout form.

W

wait state

A machine cycle in which the MPU is marking time, usually waiting for an I/O device to ready itself, or for the last instruction to be executed and a new instruction to be loaded.

warm boot

Booting a computer that has already been powered up.

Windows®

A graphical user interface from Microsoft Corporation. It uses a graphical display to represent procedures and programs that can be executed by the computer. Multiple programs can run at the same time.

word

Refers to the amount of data that can be held in a computer's registers during a process, and is considered to be the computer's basic storage unit.

word length

The major determining factor when considering the power of a given computer, because its internal hardware is chosen to conduct operations on data units composed of a specific number of bits. The larger the word length, the more data a machine can be expected to process per unit of time.

write

Any act of recording digital data, whether it be on disk, tape, or in ROM or RAM.

write gap

A slot in the R/W head across which the flux travels into the disk or tape media, from the head, during write operations.

X

x-axis

1) In a two-dimensional matrix, the horizontal row(s), such as on an oscilloscope screen.

2) The dimension of width in a graphical representation.

Xmodem

An early, and simple, asynchronous data communications protocol developed for personal computers, and capable of detecting some transfer errors, but not all.

Xon-Xoff

An asynchronous data communications protocol that provides for synchronization between the receiver and transmitter, requiring the receiver to indicate its capability to accept data by sending either an Xon (transmit on—buffer ready) or Xoff (transmit off—buffer full) signal to the transmitter.

x-y matrix

Any two-dimensional form or image, where x represents width and y represents height.

Y

y-axis

1) In a two-dimensional matrix, the vertical column(s), such as on an oscilloscope screen.

2) The dimension of height in a graphical representation.

Z

zero wait state

Describes high-speed data transfer memory operations requiring absolutely no cycle delay whatsoever.

Appendix C

A+ Objective Map

The CompTia organization has established the following objectives for the Core portion of the A+ Certification examination.

1.0 Installation, Configuration, and Upgrading

This section challenges the test taker to identify, install, configure, and upgrade microcomputer modules and peripherals. Established procedures for system assembly and disassembly must be followed. Test elements include the ability to identify and configure IRQs, DMAs, I/O addresses, and to properly set configuration switches and jumpers.

1.1 Identify basic terms, concepts, and functions of system modules, including how each module should work during normal operation. Examples of concepts and modules include the following:

- System board - Chapters 2 and 6
- Power supply - Chapter 2
- CPU/microprocessor - Chapters 1, 2, and 6
- Memory - Chapters 1, 2, and 3
- Storage devices - Chapters 1, 2, and 8
- Monitor - Chapters 1, 2, and 9
- Modem - Chapters 1 and 11
- Firmware - Chapters 1 and 3

- Boot process - Chapter 3
- BIOS - Chapters 1 and 3
- CMOS - Chapter 3

1.2 Identify basic procedures for adding and removing field replaceable modules. Examples of modules include the following:

- System board - Chapter 6, Lab Procedure 1
- Power supply - Chapter 5, Lab Procedure 1
- CPU/microprocessor - Chapters 2 and 6
- Memory - Chapter 6, Lab Procedure 12
- Storage devices - Chapter 8, Lab Procedures 1, 27, and 29
- Input devices - Chapter 7, Lab Procedure 1

1.3 Identify available IRQs, DMAs, and I/O addresses with procedures for configuring them for device installation. Examples include the following:

- Standard IRQ settings - Chapter 6
- Modems - Chapter 11
- Floppy Drives - Chapters 6 and 8
- Hard Drives - Chapters 6 and 8

1.4 Identify common peripheral ports, associated cables, and their connectors. Examples include the following:

- Cable types - Chapter 7
- Cable orientation - Lab Procedures 1, 20, 21, 26, 28, 29, 30, 32, and 33
- Serial versus parallel - Chapter 7
- Pin connections - Chapter 7

Examples of connector types include the following:

- DB-9 - Chapters 7 and 10
- DB-25 - Chapter 7
- RJ-11 - Chapter 11
- BNC-14 - Chapter 11
- RJ-45 - Chapter 11
- PS2/Mini-DIN - Chapter 7

1.5 Identify proper procedures for installing and configuring IDE/EIDE devices. Examples include the following:

- Master/slave - Chapter 8
- Devices per channel - Chapter 8

1.6 Identify proper procedures for installing and configuring SCSI devices. Topics include the following:

- Address/termination conflicts - Chapter 8
- Cabling - Chapter 8
- Types (standard, wide, fast, ultra wide) - Chapter 8
- Internal versus external - Chapter 8
- Switch and jumper settings - Chapter 8

1.7 Identify proper procedures for installing and configuring peripheral devices. Topics include the following:

- Monitor/video card - Chapter 9
- Modem - Chapter 11
- Storage devices - Chapter 8

1.8 Identify procedures for upgrading BIOS.

- Methods for upgrading - Chapter 3
- When to upgrade - Chapter 3

1.9 Identify hardware methods of system optimization and when to use them. Examples include the following:

- Memory - Chapter 6
- Hard drives - Chapter 8
- CPU/microprocessors - Chapter 6
- Cache memory - Chapter 6

2.0 Diagnosing and Troubleshooting

This item requires the test taker to apply knowledge relating to diagnosing and troubleshooting common module problems and system malfunctions. This includes knowledge of the symptoms relating to common problems.

2.1 Identify common symptoms and problems associated with each module and how to troubleshoot and isolate the problems. Contents may include the following:

- Processor/memory symptoms - Chapter 6, Lab Procedures 7 and 13
- Keyboards/mouse/trackball ball/pen/microphones/touch pad - Chapter 7, 20, and 21
- Floppy drive failures - Chapter 8, Lab Procedures 7 and 26
- Parallel ports/scanners/tape drives - Chapter 7
- Hard drives - Chapter 8, Lab Procedures 7 and 28
- Sounds (card/audio) - Chapter 12
- Monitor/video - Chapter 9, Lab Procedures 7, 22, 23, 24, and 25

- Motherboards - Chapter 6
- Modems - Chapter 11
- BIOS - Chapters 3 and 6
- CMOS - Chapter 3
- Power supply - Chapter 5, Lab Procedures 7, 17, and 18
- Slot covers - Chapter 5
- POST audible/visual error codes - Chapter 3
- Troubleshooting Tools, e.g., multimeter

2.2 Identify basic troubleshooting procedures and good practices for eliciting problem symptoms from customers. Topics include the following:

- Troubleshooting/isolation/problem determination - Chapter 5, Appendix A
- Determine whether hardware or software problem - Chapter 5

Gather information from user regarding the following:

- Customer environment - Appendix A
- Symptoms/error codes - Chapter 5
- Situation when the problem occurred - Appendix A

3.0 Safety and Preventive Maintenance

This section requires the test taker to show knowledge of safety and preventive maintenance. With regard to safety, it includes the potential hazards to personnel and equipment when working with lasers, high-voltage equipment, electrostatic discharge (ESD), and items that require special disposal procedures that comply with environmental guidelines. With regard to preventive maintenance, this includes knowledge of preventive maintenance products, procedures, environmental hazards, and precautions when working on microcomputer systems.

3.1 Identify the purpose of various types of preventive-maintenance products and procedures, and when to use/perform them. Examples include the following:

- Liquid cleaning compounds - Chapter 13
- Types of materials to clean contacts and connections - Chapter 13
- Vacuum out systems, power supplies, fans - Chapter 13

3.2 Identify procedures and devices for protecting against environmental hazards.

- UPS (uninterruptible power supply), suppressors, noise filters, and plug strips - Chapter 13
- Determining the signs of power issues - Chapter 13
- Proper methods of component storage for future use - Chapter 13

3.3 Identify the potential hazards and proper safety procedures relating to lasers and high-voltage equipment.

- Blindness caused by lasers - Chapter 13
- Electrocution caused by high-voltage equipment (for example, power supply, CRT) - Chapter 13

3.4 Identify items that require special disposal procedures that comply with environmental guidelines. Examples include the following:

- Batteries - Chapter 13
- Toner kits/cartridges - Chapter 13

- Chemical solvents and cans - Chapter 13
- CRTs - Chapter 13
- MSDS (Material Safety Data Sheet) - Chapter 13

3.5 Identify ESD (electrostatic discharge) precautions and procedures, including the use of ESD protection devices.

- What ESD can do, how it may be apparent or hidden - Chapter 13
- Common ESD protection devices - Chapter 13
- Situations that could present a danger or hazard - Chapter 13

4.0 Motherboard/Processors/Memory

This section requires the test taker to demonstrate knowledge of specific terminology, facts, ways and means of dealing with classifications, categories and principles of motherboards, processors, and memory in microcomputer systems.

4.1 Distinguish between the popular CPU chips in terms of their basic characteristics. Popular CPU chips include the following:

- Popular CPU Chips

Characteristics include the following:

- Physical size - Chapter 6
- Voltage - Chapter 6
- Speeds - Chapter 6
- Onboard cache or not - Chapter 6
- Sockets - Chapters 2 and 6
- Number of pins - Chapter 6

4.2 Identify the categories of RAM (Random Access Memory) terminology, their locations, and physical characteristics. Terminology includes the following:

- EDO RAM (Extended Data Output RAM) - Chapter 6
- DRAM (Dynamic Random Access Memory) - Chapter 6
- SRAM (Static RAM) - Chapter 6
- VRAM (Video RAM) - Chapter 6
- WRAM (Windows Accelerator Card RAM) - Chapter 6

Locations and physical characteristics:

- Memory bank - Chapters 2 and 6
- Memory chips (8-bit, 16-bit, and 32-bit) - Chapter 2
- SIMMs (Single Inline Memory Modules) - Chapters 2 and 6
- DIMMs (Dual Inline Memory Modules) - Chapters 2 and 6
- Parity chips versus nonparity chips - Chapter 6

4.3 Identify the most popular type of motherboards, their components, and their architecture (for example, bus structures and power supplies).

Types of motherboards:

- AT (Full and Baby) - Chapter 6
- ATX - Chapter 6

Motherboard components include the following:

- Communication ports - Chapter 7
- SIMM and DIMM - Chapter 2
- Processor sockets - Chapter 6

- External cache memory (Level 2) - Chapter 6
- ROM - Chapter 6

Bus architecture:

- ISA - Chapters 2 and 6
- EISA - Chapters 2 and 6
- PCI - Chapters 2 and 6
- USB (Universal Serial Bus) - Chapter 6
- VESA local bus (VL-BUS) - Chapters 2 and 6
- PC Card (PCMCIA) - Chapters 2 and 6

4.4 Identify the purpose of CMOS (Complementary Metal-Oxide Semiconductor), what it contains, and how to change its basic parameters. Examples include the folowing:

- Printer parallel port (uni/bidirectional, disable/enable, ECP, EPP) - Chapters 3 and 7
- COM/serial port (memory address, interrupt request, disable) - Chapters 3 and 7
- Hard drive (size and drive type) - Chapters 3 and 8
- Floppy drive (enable/disable drive or boot, speed, density) - Chapters 3 and 8
- Boot sequence - Chapter 3
- Memory (parity, nonparity) - Chapters 3 and 6
- Date/time - Chapter 3
- Passwords - Chapter 3

5.0 Printers

This domain requires knowledge of basic types of printers, basic concepts, printer components, how they work, how they print onto a page, paper path, care and service techniques, and common problems.

5.1 Identify basic concepts, printer operations, printer components, and field replaceable units in primary printer types.

Types of printers include the following:

- Laser - Chapter 10
- Ink-jet - Chapter 10
- Dot-matrix - Chapter 10

Paper feeder mechanisms - Chapter 10

5.2 Identify care and service techniques and common problems with primary printer types. Examples include the following:

- Feed and output - Chapter 10
- Paper jam - Chapter 10
- Print quality - Chapter 10

5.3 Identify the types of printer connections and configurations. Topics include the following:

- Parallel - Chapter 10
- Serial - Chapter 10
- Network - Chapter 10

6.0 Portable Systems

This section requires the test taker to demonstrate knowledge of portable computers and their unique components and problems.

6.1 Identify the unique components of portable systems and their unique problems. Examples include the following:

- Battery - Chapter 2
- LCD - Chapters 1, 2, and 9
- AC adapter - Chapter 2
- Docking stations - Chapter 2
- Hard drive - Chapters 1 and 2
- Types I, II, III cards - Chapters 2 and 6
- Network cards - Chapter 2
- Memory - Chapter 2

7.0 Basic Networking

This section requires the test taker to demonstrate knowledge of basic network concepts and terminology, ability to determine whether a computer is networked, knowledge of procedures for swapping and configuring network interface cards, and knowledge of the ramifications of repairs when a computer is networked.

7.1 Identify basic networking concepts, including how a network works. Examples include the following:

- Network access - Chapter 11, Lab Procedure 33
- Protocol - Chapter 11, Lab Procedure 33
- Network interface cards - Chapter 11, Lab Procedure 33
- Full duplexing - Chapter 11

- Cabling (twisted-pair, coaxial, fiber-optic) - Chapter 11
- Ways to network a PC - Chapter 11

7.2 Identify procedures for swapping and configuring network interface cards. - Chapter 11

7.3 Identify ramifications of repairs on the network. Examples include the following:

- Reduced bandwidth - Chapter 11
- Loss of data - Chapter 11
- Network slowdown - Chapter 11

8.0 Customer Satisfaction

This section requires the test taker to show knowledge of—and sensitivity around—those behaviors that contribute to satisfying customers. More specifically, these behaviors include such things as the quality of technician-customer personal interaction, the way a technician conducts himself professionally within the customer's business setting, the credibility and confidence projected by the technician (which, in turn, engenders customer confidence), and the resilience, friendliness, and efficiency that can unexpectedly delight the customer above and beyond the solving of a technical problem.

8.1 Differentiate effective from ineffective behaviors as these contribute to the maintenance or achievement of customer satisfaction. Some of the customer-satisfaction behaviors and factors addressed include the following:

- Communicating and listening (face-to face or over the phone) - Appendix A
- Interpreting verbal and nonverbal cues - Appendix A
- Responding appropriately to the customer's technical level - Appendix A

- Establishing personal rapport with the customer - Appendix A
- Professional conduct - Appendix A
- Helping and guiding customers with problem descriptions - Appendix A
- Responding to and closing a service call - Appendix A
- Showing empathy and flexibility - Appendix A
- Sharing the customer's sense of urgency - Appendix A
- Handling complaints and upset customers, conflict avoidance and resolution - Appendix A

DOS/Windows Module Examination

In addition to the Core objectives, the following objectives have been established for the DOS/Windows portion of the A+ examination.

1.0 Function, Structure, Operation, and File Management

This section requires the test taker to demonstrate knowledge of DOS, Windows 3.x, and Windows 95 operating systems in terms of functions and structure for managing files and directories and running programs. It also includes navigating through the operating system from DOS command-line prompts and Windows procedures for accessing and retrieving information.

1.1 Identify the operating system's functions, structure, and major system files. Examples include the following:

- Functions of DOS, Windows 3.x, and Windows 95 - Chapters 3 and 4
- Major components of DOS, Windows 3.x, and Windows 95 - Chapters 3 and 4
- Contrasts between Windows 3.x and Windows 95 - Chapter 4

Describe major system files: What they are, where they are located, and how they are used. Examples include the following:

- System, configuration, and user interface files - Chapters 3 and 4

DOS

- AUTOEXEC.BAT - Chapter 3
- CONFIG.SYS - Chapter 3
- IO.SYS - Chapter 3
- ANSI.SYS - Chapter 3
- MSDOS.SYS - Chapter 3
- EMM386.EXE - Chapter 3
- HIMEM.SYS - Chapter 3
- COMMAND.COM - Chapter 3

Windows 3.x

- WIN.INI - Chapter 4
- SYSTEM.INI - Chapter 4
- USER.EXE - Chapter 4
- GDI.EXE - Chapter 4
- WIN.INI - Chapter 4
- WIN.COM - Chapter 4
- PROGMAN.INI - Chapter 4
- PROGMAN.EXE - Chapter 4
- KRNLXXX.EXE - Chapter 4

Windows 95

- IO.SYS - Chapter 4
- MSDOS.SYS - Chapter 4
- COMMAND.COM - Chapter 4
- REGEDIT.EXE - Chapter 4
- SYSTEM.DAT - Chapter 4
- USER.DAT - Chapter 4

1.2 Identify ways to navigate the operating system and how to get to needed technical information.

Procedures (menu or icon-driven) for navigating through DOS to perform such things as locating, accessing, and retrieving information - Chapter 3

Procedures for navigating through the Windows 3.x/Windows 95 operating system, accessing, and retrieving information - Chapter 4, Lab Procedures 3, 4, 5, and 6

1.3 Identify basic concepts and procedures for creating and managing files and directories in DOS/Windows. Examples include the following:

- File attributes - Chapter 3
- File naming conventions - Chapter 3, Lab Procedure 2
- File types, file formats - Chapter 3, Lab Procedure 2
- Command syntax - Chapter 3, Lab Procedure 2
- Read Only, Hidden, System, and Archive attributes - Chapter 3

1.4 Identify the procedures for basic disk management. Examples include the following:

- Using disk management utilities - Chapter 13, Lab Procedure 36
- Backing up - Chapter 13, Lab Procedure 36
- Formatting - Chapter 8, Lab Procedures 27, 28, and 37
- Partitioning - Chapter 8, Lab Procedures 27, 28, and 37
- Defragmenting - Chapter 13, Lab Procedure 36
- ScanDisk - Chapter 13
- FAT 32 - Chapter 4
- File Attention Tables (FAT) - Chapter 4
- Virtual File Allocation Tables (VFAT) - Chapter 4

2.0 Memory Management

This section requires the test taker to demonstrate knowledge of the types of memory used by DOS and Windows, and the potential for memory address conflicts.

2.1 Differentiate between types of memory. Examples include the following:

- Conventional - Chapter 3
- Extended/upper memory - Chapter 3
- High memory - Chapter 3
- Expanded memory - Chapter 3
- Virtual memory - Chapter 3

2.2 Identify typical memory-conflict problems and how to optimize memory use. Examples include:

- What a memory conflict is - Chapter 5
- How it happens - Chapter 5
- When to employ utilities - Chapter 5
- System monitor - Chapter 5
- General Protection Fault - Chapter 5
- Illegal operations occurrences - Chapter 5
- MemMaker or other optimization utilities - Chapters 3, 5, and 6, Lab Procedure 12
- HIMEM.SYS - Chapters 3 and 5
- SMARTDRV - Chapter 3
- Use of expanded memory blocks (using EMM386.EXE) - Chapter 3

3.0 Installation, Configuration, and Upgrading

This section requires the test taker to demonstrate knowledge of installation, configuration, and upgrading DOS, Windows 3.x, and Windows 95. This includes knowledge of system boot sequences.

3.1 Identify the procedures for installing DOS, Windows 3.x, Windows 95, and bringing the software to a basic operational level. Examples include the following:

- Partition - Chapter 8, Lab Procedures 27, 28, and 37
- Format drive - Chapter 8, Lab Procedures 27, 28, and 37
- Run appropriate setup utility - Chapter 8
- Loading drivers - Chapter 8

3.2 Identify steps to perform an operating system upgrade. Topics include the following:

- Upgrading from DOS to Windows 95 - Chapter 4
- Upgrading from Windows 3.x to Windows 95 - Chapter 4, Lab Procedure 5

3.3 Identify the basic system boot sequences and alternative ways to boot the system software, including steps to create an emergency boot disk with utilities installed. Examples include the following:

- Files required to boot - Chapters 3 and 4
- Creating emergency boot disks - Chapter 4
- Startup disk - Chapter 4
- Safe mode - Chapters 4 and 5
- DOS mode - Chapter 4

3.4 Identify procedures for loading/adding device drivers and the necessary software for certain devices.

DOS

- Windows 3.x procedures - Chapter 3
- Windows 95 Plug and Play - Chapters 3 and 4

3.5 Identify the procedures for changing options, configuring, and using the Windows printing subsystem. - Chapter 10, Lab Procedure 30

3.6 Identify the procedures for installing and launching typical Windows and non-Windows applications. - Chapter 4

4.0 Diagnosing and Troubleshooting

This section requires the test taker to demonstrate the ability to apply knowledge to diagnose and troubleshoot common problems relating to DOS, Windows 3.x, and Windows 95. This includes understanding normal operation and symptoms relating to common problems.

4.1 Recognize and interpret the meaning of common error codes, and startup messages from the boot sequence, and identify steps to correct the problem.

- Incorrect DOS version - Chapter 5
- Error in CONFIG.SYS line *xx* - Chapter 5
- Bad or missing COMMAND.COM - Chapter 5
- HIMEM.SYS not loaded - Chapter 5
- Swap file corrupt - Chapter 5
- A device referenced in WIN.INI could not be found. - Chapter 5
- Missing or corrupt HIMEM.SYS - Chapter 5
- No operating system found - Chapter 5
- Safe mode - Chapter 5

4.2 Recognize Windows-specific printing problems and identify the procedures for correcting them. Examples include the following:

- Print spool is stalled. - Chapter 10
- Incorrect/incompatible driver for print - Chapter 10

4.3 Recognize common problems and determine how to resolve them. Topics include the following:

- General protection faults - Chapter 5
- Illegal operation - Chapter 5
- Invalid working directory - Chapter 5
- System lockup - Chapter 5
- Option will not function. - Chapter 5
- Application will not start or load. - Chapter 5
- Cannot log on to network - Chapter 11

DOS and Windows-based utilities:

- MSD.EXE - Chapter 5, Lab Procedure 8
- ScanDisk - Chapter 13, Lab Procedure 36
- DEFRAG.EXE - Chapter 13, Lab Procedure 36
- MEM.EXE - Chapter 3, Lab Procedure 12
- EDIT.COM - Chapters 3 and 5
- FDISK.EXE - Chapter 8, Lab Procedure 27
- ATTRIB.EXE - Chapter 3
- Device Manager - Chapter 5
- SYSEDIT.EXE - Chapters 4 and 5, Lab Procedure 6
- EXTRACT.EXE - Chapter 5

4.4 Identify concepts relating to viruses and virus types—their danger, their symptoms, sources of viruses, how they infect, how to protect against them, and how to identify/remove them. Topics include the following:

- What they are - Chapter 13
- Sources - Chapter 13
- How to determine their presence - Chapter 13

5.0 Networks

This section requires the test taker to demonstrate knowledge of the network capabilities of DOS and Windows and how to connect to networks, including what the Internet is about, its capabilities, basic concepts relating to Internet access, and generic procedures for system setup.

5.1 Identify the networking capabilities of DOS and Windows, including procedures for connecting to the network. Topics include the following:

- Sharing disk drives - Chapter 11, Lab Procedure 33
- Sharing print and file services - Chapter 11, Lab Procedure 33
- Network type and network card - Chapter 11

5.2 Identify concepts and capabilities relating to the Internet and basic procedures for setting up a system for Internet access. Topics include the following:

- TCP/IP - Chapter 11
- E-mail - Chapter 11
- HTML - Chapter 11
- HTTP:// - Chapter 11
- FTP - Chapter 11
- Domain names (Web sites) - Chapter 11
- ISP - Chapter 11
- Dial-up access - Chapter 11

Index

Symbols

A

B

C

D

E

F

G

H

I

J - K

L

M

N

O

P

Q - R

S

T

U

V

W

X - Y - Z

What's on the CD-ROM

To run the CD-ROM: Most Windows 95 and Windows NT machines should automatically run the CD-ROM when it is inserted into its drive. When this is not the case, you can go to the root of your CD-ROM drive (through Windows Explorer, for example), and run the file SETUP.EXE. Regretfully, we cannot offer a version of this CD-ROM that runs in the Macintosh environment at this time. The CD-ROM features the following contents.

The **TestPrep application suite** contains preparation material for A+ certification. There are actually three TestPrep executables on the CD-ROM. Each uses a database of sample test questions by Marcraft International, with whose cooperation this book was produced.

Study cards are electronic questions-and-answer scenarios that help you prepare for the simulation test -- or the real thing. Fire up the study cards and you can run through test questions at random, testing your knowledge and learning your strong points and weaker points. Then focus your studies even more.

Flash cards are a very simple-yet-effective way to test your retention of the material. They work just like the old reliable flash-card method you may have used in school at some point. Run the program and a question will appear onscreen; think about it; if you're not sure of the answer, look up the subject area in the book (electronically or with the hard copy of the book) and review it quickly; then click on the answer button.

The **Practice Exam** helps you learn the test. The Flash Cards and Study Cards help you learn the technology. You'll be timed, scored, showing in detail how you did, and given the means to check your answers so that you can study what you missed.

Unlike the Flash Cards and the Study Cards applications, when you use the practice exam component, you will not be able to link to the electronic book content included on the CD-ROM. You also will not be able to reveal answers to questions while taking the test. This is simply to better duplicate the actual exam-taking scenario you'll face when you go in for the real test.

The 150-Question Version. Running the file \TESTPREP\TPMENU.EXE starts up the version with 150 test questions available for free. You'll get the same 150 questions each time you take this exam, but they'll be compiled in random order each time.

FREE COUPON OFFER

Running the 800-Question Version. \MARCRAFT\MARCRAFT.EXE. is the version with over **800 more questions** available from Marcraft for $50. Just follow the directions on the CD-ROM for completing the transaction with Marcraft. You will receive a password for unlocking the complete database of questions...also a coupon good for **$50 off your A+ examination**, making the extra test questions **FREE**. TURN 1 PAGE BACK or call 1-800-441-6006

The Linked Electronic Edition of A+ Certification Training Guide. The entire book you hold in your hands now -- save for the front matter, index, and end matter -- is on this CD-ROM in Adobe Acrobat Reader .PDF format (Acrobat Reader is included on the CD-ROM). What's more, each of the questions in the database is linked to the relative content in the book. If you find yourself at a Study Card or Flash Card question for which you need to review, simply click on the link and you'll automatically go to that portion of the book. It's that easy.

Technical support issues: If you need assistance with how this CD operates, please access the information available on our website at http://www.mcp.com/feedback. Our most Frequently Asked Questions are answered there. If you do not find the answers to your questions on our website, you may contact Macmillan User Services at (317) 581-3833, or e-mail us at support@mcp.com. If you have questions about any of the practice questions, please contact Marcraft at (800)441–6006.

Charles J. Brooks,
Marcraft International Corporation

A+ Certification

TRAINING GUIDE

New Riders

A+ Certification Training Guide

Library of Congress Catalog No.: 98-85227

ISBN: 1-56205-896-7

01 00 99 98 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 98-1 shows that the first printing of the book occurred in 1998.

Screen reproductions in this book were created using Collage Plus from Inner Media, Inc., Hollis, NH.

Executive Editor
John Kane

Acquisitions Editor
Steve Weiss

Development Editor
Nancy Warner

Managing Editor
Sarah Kearns

Project Editor
Mike La Bonne

Copy Editor
Keith Cline

Indexer
Cheryl A. Jackson

Technical Editors
Brian Alley
Melissa Ryan

Software Specialists
Jack Belbot
Todd Pfeffer

Production Team
Steve Balle-Gifford
Maribeth Echard
Mary Hunt

About the Author

Charles J. Brooks is currently vice president in charge of research and development at Marcraft International Corporation, Pasco, WA. He is the author of several books, including *Speech Synthesis*, *Pneumatic Instrumentation*, *The Complete Introductory Computer Course*, *Radio-Controlled Car Project Manual*, and *IBM PC Peripheral Troubleshooting & Repair*. A former electronics instructor and technical writer with the National Education Corporation, Charles taught and wrote on post-secondary EET curriculum, including introductory electronics, transistor theory, linear integrated circuits, basic digital theory, industrial electronics, microprocessors, and computer peripherals.

Contents at a Glance

Table of Contents

We'd Like to Hear from You!

As part of our continuing effort to produce books of the highest possible quality, MCP would like to hear your comments. To stay competitive, we really want you, as a computer book reader and user, to let us know what you like or dislike most about this book or other Macmillan products.

You can send comments, ideas, or suggestions for improving future editions to us at **certification@mcp.com**. The address of our Internet site is **http://www.mcp.com** (World Wide Web).

Thanks in advance—your comments help us to continue publishing the best books available on computer topics in today's market.

Note

Although we cannot provide general technical support, we're happy to help you resolve problems you encounter related to our books, disks, or other products. If you need such assistance, please contact our Tech Support department at 800-545-5914 ext. 3833.