Chapter 17

System Upgrades and Improvements

You want your PC to stay viable—even powerful—in a world of bigger, better, and faster machines. But how fast is fast?

- Do your software programs load quickly?
- Are you frustrated by waiting each time you save information to your hard disk?
- Does the cursor move when you want it to, where you want it to, as fast as you want it to?
- Do you find yourself watching the disk-drive lights as you wait?

Your computer's performance is measured by getting the most work done in as little time as possible. When your computer is slow, you have three options: stay with your current computer as is, upgrade it, or buy a new PC. Most often, upgrading is the preferred choice. Why? For several reasons:

- You can preserve your investment in a paid-for computer.
- You can expand your knowledge of computers by upgrading the PC yourself.
- You can save money by repairing and upgrading your computer until you can afford a new computer.

You can extend the life of older systems by adding functions and features that nearly match those of the newest systems on the market. You also can add new components to a modern system to squeeze cutting-edge performance out of what already is a premium system.

In this chapter, you learn about these types of system upgrades:

- Increasing system memory
- Upgrading a ROM BIOS

- Increasing disk storage
- Increasing system speed
- Improving a video subsystem
- Upgrading to a newer version of DOS

Caution

Avoid making radical upgrades. Sometimes, people throw far too much money and materials into an archaic system. You could upgrade a typical PC- or XT-class system to a 486 AT-class system, for example, but the result may be an expensive and lopsided system. The lightning speed of the processor would be slowed by the limping hard disk, and some expansion cards may not work with the new processor. You could spend hundreds of dollars to replace almost everything in an existing system, which may be more expensive than purchasing a new system. By making extensive changes in a system, you risk creating a Frankenstein system with strange quirks and eccentric—if not dangerous—operation. This type of system might be acceptable for personal use, but it is not recommended for use as a business system. In business, you have no time to coerce a reluctant "business partner."

Upgrading Goals

IBM and IBM-compatible systems are easy to upgrade and improve, not only because of support from IBM, but also because of the huge industry that has grown up around the IBM-compatible computer standard. For example, third-party manufacturers not only produce a wide range of complete IBM-compatible systems, but also make virtually every component for what are known as IBM *clones*. IBM uses many third-party components in its own systems as well. Because of the third-party support, a wide range of options is available for upgrading a system to achieve increased performance or expanded capabilities.

Although you can make many improvements to your PC, most of those improvements require you to open your computer. This situation is not unnatural; the first PC, unveiled in 1981, included five internal expansion slots for future enhancements. By buying various expansion cards for these slots, you can add more power and features to your PC. Also, you must open the system unit to add more memory, a new ROM BIOS chip, a new hard drive, and other items.

You should have a strategy for upgrading your PC. Upgrading isn't simply a matter of getting your PC to work faster and more reliably. When you travel, you can elect to take a bus, a car, a plane, or the subway; the decision depends on how far you have to go, how much money you want to spend, and how quickly you want to get there. The same is true of upgrading.

Consider what your PC keeps you from doing now. Is the screen too small to use for extended periods? Does the hard disk occasionally lose information? Is your machine not able to run the shareware programs that your brother-in-law passes on to you? Turn these weaknesses into goals that can be measured later with each improvement. If your goal can't be measured, you'll never know whether you achieved it.

Next, consider whether you can afford the upgrade today. If so, order right away. If not, you have the following choices:

- Wait to upgrade until you can afford it.
- Purchase another upgrade that is more in your financial ballpark. For example, you may forgo a new hard disk and instead get some extra memory, which costs less.
- Limit the features of your upgrade so that it is affordable. You can order a smaller hard disk, for example, and still benefit from its high speed.

Upgrading by Increasing System Memory

Adding memory to a system is one of the most useful upgrades that you can perform and also one of the least expensive, especially when you consider the increased capabilities of DOS, Microsoft Windows, and OS/2 when you give them access to more memory. In some cases, doubling the memory can virtually double the speed of a computer.

Memory chips come in different shapes and sizes, yet all memory chips in which you are interested are called *DRAM*, or dynamic random-access memory. DRAM chips are the most common type of memory. These chips are considered to be dynamic because they need to be energized hundreds of times per second to hold information. If you shut off the power, the information is lost.

You can add three primary types of memory to a system: conventional, extended, and expanded. (These types of memory are described in detail in Chapter 2, "Overview of System Features and Components," and Chapter 7, "Memory.")

This section discusses adding memory, including selecting memory chips, installing memory chips, and testing the installation.

Conventional Memory

Conventional memory is the memory within the first megabyte of a system's memory that can be used by DOS or application software. Conventional memory is accessible in the native mode of 8088 processors and in the real mode of operation of 80286 and higher processors. Today, virtually all systems are sold with the full 640K of usable conventional memory installed, if not more.

At this writing, 4M (4,096K) of memory is the bare minimum offered with most systems. If you have an older system and do not have the full complement of 640K conventional memory, you can add extended or expanded memory only after filling the 640K conventional-memory space.

Extended Memory

Extended memory is memory above 1M. This memory usually is not available to your computer directly, except through special software programs or drivers. This memory is available only in 80286 and higher computers.

PC and XT systems cannot use extended memory, because the 8088/8086 processors in PC and XT systems cannot run in protected mode. Expanded memory can enhance some of the capabilities—but generally not the speed—of these computer systems.

You can install both extended and expanded memory in AT-class systems. Extended memory is more useful than expanded memory. Windows and OS/2 can take full advantage of the 16M of available conventional- and extended-memory space. The need or desire for expanded memory in AT systems is virtually nonexistent because this expanded memory must be implemented in a slow and clumsy way. Fortunately, you easily can change almost any AT-type, 16-bit expanded memory board from expanded to extended memory by reconfiguring the adapter switches.

Expanded Memory

Expanded memory basically is converted extended memory. Introduced in 1985, expanded memory relies on a technical trick known as *paging* or *bank-switching*. Paging involves accessing large amounts of memory by swapping it, in 16K chunks, into and out of a 64K memory window. This window is placed in the unused memory space between 640K and 1M to access memory up to 32M.

Expanded memory is slow and clumsy for the system to use and generally is considered to be obsolete by today's standards. 80386 and higher systems can create expanded memory from extended memory, so normally, you only physically add extended memory to a system. Also, DOS, Windows, and OS/2 can use extended memory directly, but not expanded memory.

IBM and Microsoft have added limited expanded-memory support to DOS, starting with DOS 4.0. Expanded memory requires specialized hardware to implement the bank-switching techniques, so for PC and XT systems, you must purchase a special type of memory adapter that has built-in switching capability. If you decide to add one or more of these expanded-memory adapters to a PC- or XT-type system, be sure that the adapter meets Expanded Memory Specification 4.0 (EMS 4.0) or higher standards in the hardware and the software. Many older expanded-memory adapters meet only the inferior EMS 3.x standards in hardware, even though the software driver may comply with EMS 4.x. The major difference between the 3.x and 4.x EMS designs is that the newer 4.x versions provide a larger, variable-size paging area that can occupy a larger range of memory areas within the system memory map.

Because they use advanced AT chipsets, some 286 AT-class systems can use extended memory to emulate expanded memory. If your 286 system has this capability, check your system documentation for more information. An enhanced Setup program usually controls this capability.

Upgrade Strategies

Adding memory can be an inexpensive solution; at this writing, the cost of memory is about $45 per megabyte. A small dose can give your computer's performance a big boost.

How do you add memory to your PC? You have three options, listed in order of convenience and cost:

- Adding memory in vacant slots on your motherboard
- Replacing your current motherboard's memory with higher-capacity memory
- Purchasing a memory expansion card

Adding expanded memory to PC- or XT-type systems is not a good idea, mainly because an expanded memory board with a couple of megabytes of expanded memory installed can cost more than the entire system is worth. Also, this memory does not function for Windows, and a PC- or XT-class system cannot run OS/2. Instead, purchase a more powerful system—for example, an inexpensive 486SX—with greater expansion capabilities.

If you decide to upgrade to a more powerful computer system, you normally cannot salvage the memory from a PC or XT system. The 8-bit memory boards are useless in AT and Micro Channel systems, and the speed of the memory chips usually is inadequate for newer systems. Many new systems use high-speed SIMM modules rather than chips. A pile of 150-ns (nanoseconds), 64K or 256K chips is useless if your next system is a high-speed system that uses SIMMs or memory devices faster than 70 ns.

Be sure to weigh carefully your future needs for computing speed and for a multitasking operating system (OS/2, for example) with the amount of money that you spend to upgrade current equipment.

Adding Motherboard Memory

This section discusses *motherboard memory*—the memory actually installed on the motherboard—rather than the memory that resides on adapter boards. The first part of this section presents recommendations on selecting and installing chips. The last part of the section provides instructions for modifying an IBM XT Type 1 motherboard. This modification enables you to place a full 640K of memory on the motherboard, eliminating the need for memory-expansion boards. IBM's more recent XT Type 2 motherboards already include this modification.

Selecting and Installing Memory Chips or SIMMs

If you are upgrading a motherboard by adding memory, follow the manufacturer's guidelines on which memory chips or modules to purchase. As you learned in Chapter 7, "Memory," memory comes in various form factors, including individual chips known as DIP (dual in-line pin) memory chips, SIMMs (single in-line memory modules), and SIPPs (single in-line pin packages). Your computer may use one or possibly a mixture of these form factors.

The maker of your computer's motherboard determines what type of memory chips are used. The following list describes each chip or module type:

- *DIPs*. Early computers used DIP (dual in-line pin) memory chips. A DIP memory chip is a rectangular chip that has 16 metal legs, eight on each side. To install such a memory chip, you must plug it in place. DIP chips are installed in multiples of nine. For example, you must install 36 separate 256-kilobit chips to acquire 1M of memory. Sometimes, the DIPs are permanently soldered to your motherboard.

- *SIMMs*. Single in-line memory modules are like small circuit boards with chips soldered on them. Different numbers of chips can be mounted on the SIMM, and the chips can be mounted on one or both sides of the SIMM. A SIMM has a row of contacts on one edge of the board. The contacts can be tin- or gold-plated. SIMMs are retained in the system by special sockets with positive latching mechanisms that lock the SIMMs in place. SIMM connectors use a high-force wiping contact that is extremely resistant to corrosion.

  SIMMs are available in two types: 30-pin and 72-pin. The 30-pin modules come in 9-bit form with parity or 8-bit form for systems that lack parity checking. The 72-pin SIMMs are 36 bits wide with parity (32 data bits and 4 parity bits), or 32 bits wide without parity. Notice that the 9-bit and 36-bit SIMMs with parity always can be used in systems that lack parity checking and that the nonparity SIMMs cannot be used in normal systems that require parity bits. Systems that lack parity checking for memory are not recommended.

- *SIPPs*. Single in-line pin packages, sometimes called SIP, really are SIMMs with pins rather than contacts. The pins are designed to be installed in a long connector socket that is much cheaper than the standard SIMM socket. SIPPs are inferior to SIMMs because they lack the positive latching mechanism that retains the module, and the connector lacks the high-force wiping contacts that resist corrosion. SIPPs are rarely used today. For more information on the form factor of RAM chips, see Chapter 7, "Memory."

Whichever type of memory chips you have, the chips are installed in memory banks. A *memory bank* is a collection of memory chips that make up a block of memory. Each bank of memory is read by your processor in one pass. A memory bank does not work unless it is filled with memory chips.

Early 8088, 8086, and 80286 PCs have four memory banks that are nine chips wide. These memory banks may be labeled 0 through 3 or A through D. The first eight chips are for holding data. The ninth chip is a parity chip meant to ensure that the numbers of the other eight are correct. Each chip is measured in bits, and eight chips make up a byte. Imagine that you had the first two banks full of 256-kilobit chips. How much memory is that? Well, 18 chips (9 chips × 2 banks) of 256-kilobit memory equals 512K.

More modern 80286 computers usually can take four banks of 256-kilobit chips to make 1,024K (1M). Newer 80286 computers can handle up to 4M on the motherboard, using 1M chips. In 80386SX-based computers, four memory banks are used, requiring 18 chips, or sometimes two 9-bit SIMMs each. 80386DX, 80486SX, and 80486DX computers often have two memory banks, each using four 9-bit SIMMs or one 36-bit SIMM. Pentium computers usually have two memory banks using two 36-bit SIMMs.

Installing extra memory on your motherboard is the least expensive way to add memory to your computer. You may have a vacant memory bank in which you can install extra memory and use it to speed your computer.

RAM-Chip Type (Capacity). Individual RAM chips come in different capacities. The capacity determines the number of data bits that can be stored in a chip of a particular size. For example, RAM chips for the original IBM PC store 16 kilobits of data; these RAM chips are the smallest used in any IBM-compatible system. The RAM chips for the original version of the IBM XT store 64 kilobits of data. The standard chip for 386-based systems is the 1M chip (usually found in SIMMs). Today, 4- or 16-megabit chips (again, on SIMMs) are common in 486 and Pentium systems.

Before you add RAM to a system (or replace defective RAM chips), you must determine the memory chips required for your system. Your system documentation contains this information.

If you need to replace a defective RAM chip and do not have the system documentation, you can determine the correct chip for your system by inspecting the chips that are already installed. Each chip has markings that indicate the chip's capacity and speed. The following table lists the markings on individual 1M chips produced by various companies.

Markings	Manufacturer
TMS4C1024N/DJ	Texas Instruments
HM511000AP/AJP/AZP	Hitachi
MB81C1000P/PJ/PSZ	Fujitsu

If you do not have the documentation for your system and the manufacturer does not offer technical support, open your system case and carefully write down the markings that appear on your memory chips. Then contact a local computer store or mail-order chip vendor for help in determining the proper RAM chips for your system. Adding the wrong RAM chips to a system can make it as unreliable as leaving a defective chip on the motherboard and trying to use the system in that condition.

RAM-Chip Speed. RAM chips also come in various speeds. For example, 120-ns chips are used in older systems, and 60- or 70-ns chips are used in fast 486- and Pentium-based systems.

The motherboard manufacturer determines the correct speed of the memory chips installed in each system. IBM, for example, specifies different speed memory for different systems. Table 17.1 lists the required RAM-chip speeds and wait states for IBM motherboards.

Table 17.1 IBM Motherboard Memory Timing

System	CPU	Clock Speed (MHz)	Wait States	Memory-Access Time (ns)	Notes
PC	8088	4.77	1	200	
XT	8088	4.77	1	200	
AT	286	6	1	150	
AT	286	8	1	150	
XT-286	286	6	0	150	Zero wait
PS/1	286	10	1	120	
25	8086	8	0	150	Zero wait
30	8086	8	0	150	Zero wait
25-286	286	10	1	120	
30-286	286	10	1	120	
35 SX	386SX	20	0-2	85	Paged memory
40 SX	386SX	20	0-2	85	Paged memory
L40	386SX	20	0-2	80	Paged memory
50	286	10	1	150	
50Z	286	10	0	85	Zero wait
53 486SLC2	486SLC2	50	0-2	70	Interleaved memory, internal 16K cache
55 SX	386SX	16	0-2	100	Paged memory
56 486SLC3	486SLC3	75	0-2	70	Interleaved memory, internal 16K cache
57 SX	386SX	20	0-2	70	Paged memory
57 486SLC3	486SLC3	75	0-2	70	Interleaved memory, internal 16K cache
60	286	10	1	150	
65	386SX	16	0-2	100	Paged memory
70	386DX	16	0-2	85	Paged memory
70	386DX	20	0-2	85	Paged memory
70	386DX	25	0-5	80	External 64K cache
70	486DX	25	0-5	80	Internal 8K cache
P70	386DX	16	0-2	85	Paged memory
P70	386DX	20	0-2	85	Paged memory
P75	486DX	33	0-5	70	Internal 8K cache
76	486DX2	66	0-2	70	Interleaved memory, internal 8K cache
76	486SX	33	0-2	70	Interleaved memory, internal 8K cache

System	CPU	Clock Speed (MHz)	Wait States	Memory-Access Time (ns)	Notes
76	486DX4	100	0-2	70	Interleaved memory, internal 16K cache
77	486DX2	66	0-2	70	Interleaved memory, internal 8K cache
77	486SX	33	0-2	70	Interleaved memory, internal 8K cache
77	486DX4	100	0-2	70	Interleaved memory, internal 16K cache
80	386DX	16	0-2	80	Paged memory
80	386DX	20	0-2	80	Paged memory
80	386DX	25	0-5	80	External 64K cache
90	486SX	20	0-5	70	Interleaved memory, internal 8K cache
90	486SX	25	0-5	70	Interleaved memory, internal 8K cache
90	486DX	25	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache
90	486DX	33	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache
90	486DX	50	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache
95	486SX	20	0-5	70	Interleaved memory, internal 8K cache
95	486SX	25	0-5	70	Interleaved memory, internal 8K cache
95	486DX	25	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache
95	486DX	33	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache
95	486DX	50	0-5	70	Interleaved memory, internal 8K cache, optional external 256K cache

It won't hurt to install chips that are faster than required for your motherboard or memory card; buying faster memory chips can be good if you intend to transplant them to a faster computer in the future. Unfortunately, the faster memory won't speed your computer; your computer's design anticipates working at a certain speed and no faster.

The speed of a memory chip is printed on the surface of the chip. On the memory chips—whether they are the DIP, SIPP, or SIMM type—you will find an identifying number. The last two digits after the dash (—) are especially important, because they indicate the speed of your memory.

In some cases, you can use slower-speed memory on an adapter than would normally be tolerated by the system motherboard. Many systems run the expansion slots at a fixed slower speed (8 MHz for most ISA bus systems) so that installed adapters function properly. The PS/2 system memory adapters may be capable of running more slowly than main memory because of the Micro Channel Architecture (MCA) interface's higher level of controls and capabilities. The MCA's asynchronous design enables adapters to remain independent of the processor's speed and to request additional wait states, as required, to accommodate the slower adapters. (A *wait state* is an idle cycle in which the processor and system essentially wait for a component, such as memory, to "catch up.")

All the memory adapters that IBM offers for the PS/2 50 and 60, for example, use 120-ns memory chips. These adapters also are specified for the Model 50Z, which normally runs with no wait states and requires 85-ns memory on the motherboard. If these slower 120-ns memory adapters are used in the Model 50Z, they cause the MCA in the 50Z to insert a wait state when the system accesses the adapter memory. Only the motherboard memory, therefore, runs with zero wait states. Several third-party board manufacturers offer special zero-wait-state boards for the 50Z, which requires memory faster than 120 ns.

In some systems, the motherboard memory speed can be controlled. Systems with adjustable wait-state settings enable you to choose optimal performance by purchasing the proper high-speed memory or to choose lower performance by purchasing cheaper memory. Many compatibles offer a wait-state jumper or configuration option, which controls whether the motherboard runs with wait states. Running with zero wait states may require faster-access-speed memory.

Other systems can configure themselves dynamically to the memory installed. The PS/2 Model 90 and 95 system boards check the speeds of the SIMMs installed on the system board and adjust the number of wait states accordingly. Most other PS/2 systems simply check the speed of SIMMs installed on the system board and flag an error condition if the minimum speed requirements are not met.

Systems that use 36-bit SIMMs can detect both the speed and capacity of the installed SIMMs through four special contacts called *presence detect pins*. The motherboard can use these pins to determine the installed SIMM's rated speed and capacity, in much the same way that many cameras can tell what speed film you have loaded by "reading" a series of contacts on the film canister. In the Model 90 and Model 95, if the SIMM is slower than 70 ns, the system adds wait states so that the rest of the memory can keep up. In other systems, such as the Model 70, if memory slower than the required 80 or 85 ns is installed, a 225 POST error message appears. If you look up this error code in the IBM error-code list in Appendix A, it says `Wrong-speed memory on system board`. This error message means that the installed memory is too slow for the system.

Systems with 16 MHz or faster clock speeds require extremely fast memory to keep up with the processor. In fact, the speeds that are required are so excessive that standard DRAM typically would be replaced by faster and more expensive static RAM (SRAM). One alternative—adding wait states to reduce the memory-speed requirements—greatly decreases performance, which is not what you want in a fast system.

RAM-Chip Architecture. Some special memory-architecture schemes have been devised to reduce the number of wait states required, boost overall system performance, and keep costs down. The following architecture schemes are most commonly used to increase memory performance:

- Paged memory
- Interleaved memory
- Memory caching

Paged memory is a simple scheme for improving memory performance that divides memory into pages ranging from 512 bytes to a few kilobytes long. The paging circuitry then enables memory locations within a page to be accessed with zero wait states. If the desired memory location is outside the current page, one or more wait states are added while the system selects the new page.

Paged memory has become common in higher-end 286 systems as well as in many 386 systems. For example, many PS/2 systems, such as the Model 70 and Model 80, use paged memory to increase performance so that slower 80- or 85-ns memory can be used.

Interleaved memory offers greater performance than paged memory. This higher-performance scheme combines two banks of memory into one, organized as even and odd bytes. With this combination, an access cycle can begin in the second bank while the first is already processing a previous access, and vice versa. By alternating access to even and odd banks, you can request data from one bank, and while the request is pending, the system can move to the next bank to process another request. The first request becomes available while the second request is still pending, and so on. By interleaving access to memory in this manner, a system can effectively double memory-access performance without using faster memory chips.

Many of the highest-performance systems use interleaved memory to achieve increased performance. Some systems that offer interleaved memory can use the interleaving function only if you install banks in matched-capacity pairs, which usually means adding two 36-bit SIMMs of equal capacity at a time. If you add only a single bank or add two banks of different capacity, the system still functions, but memory interleaving is disabled, and you pay a considerable performance penalty. Consult your system's technical-reference manual for more information.

Memory caching is the most popular and usually the most effective scheme for improving memory performance. This technique relies on a small amount (8K to 512K) of raw, high-speed memory fast enough to keep up with the processor with zero wait states. This small bank of cache memory often is rated at *15 ns or less* in access speed. Because this rate is faster than normal DRAM components can handle, a special type of memory is used. This memory is called *SRAM* (static RAM). SRAM devices do not need the constant refresh signals that DRAM devices require. This feature, combined with other design properties, results in extremely fast access times and very high costs.

Although SRAM chips are expensive, only a small number of SRAM chips are required in a caching scheme. SRAM is used by a special cache-controller circuit that stores frequently accessed RAM locations and also is preloaded with the RAM values that the cache controller expects to be accessed next. The cache acts as an intelligent buffer between the CPU and slower dynamic RAM.

A *cache hit* means that the particular data the CPU wanted was available in the cache RAM and that no additional wait states are available for retrieving this data. A *cache miss* means that the data the CPU wanted had not been loaded into the cache RAM and that wait states must be inserted while the data is retrieved. A good cache controller has a hit ratio of 95 percent or more (the system runs with zero wait states 95 percent of the time). The net effect is that the system acts as though nearly all the memory is 15 ns or less in speed, although most of the memory really is much slower (and, therefore, much less costly).

Systems based on the 486SX, SL, or DX processors include a cache controller and 8K of internal-cache RAM in the CPU that makes them much faster than earlier systems. The 486SLC CPU has a 1K internal cache, whereas the Pentium CPU includes two 8K internal caches: one for code and the other for data. 486DX4 processors have a 16K internal cache, while both the 486SLC2 and 486SLC3 processors have a 16K cache. Systems with 386SX or DX processors must use an external-cache controller with externally provided cache RAM; these systems have no internal cache. The 386SL provides a built-in cache controller, and the IBM-designed 386SLC incorporates virtually the same cache controller and 8K of built-in cache RAM as the 486 processors. IBM, therefore, can claim an 80 percent performance increase from the regular 386SX or DX chips for systems that use the 386SLC processor.

A CPU internal cache is called a primary or Level 1 (L1) cache, and an external cache is called a secondary or Level 2 (L2) cache. Typically, the larger the memory cache, the better the performance. A larger secondary processor cache, however, is no guarantee; you may find that the system with the least cache RAM can outperform a system with a greater amount of cache RAM. Actual performance depends on the efficiency of the cache controller and the system design; a cache integrated into a CPU, for example, can far outperform an external cache. For example, adding the 256K L2 cache RAM option to a PS/2 Model 90 or 95 with a 486DX processor offers only a small increase in performance relative to the 8K of L1 cache memory built into the 486 CPU chip, because the L1 cache integrated into the CPU can outperform an external (L2) cache. Also, adding cache RAM does not result in a proportional increase in performance. You often gain the best performance by using the middle amount of secondary cache that your computer can accept. A PC that can accommodate a 64K, 128K, or 256K secondary cache provides the most bang for the buck.

To get maximum system performance and reliability, the best recommendation for adding chips or SIMMs to a motherboard is to use memory rated at the speeds recommended by the manufacturer. Faster memory is likely to work in the system, but it creates no performance benefit and therefore is a waste of money.

The minimum access-time specification for motherboard memory in a specific system is listed in the system's technical-reference manual. If you have an IBM-compatible system that lacks proper documentation, you can refer to table 17.1 as a guide, because most compatibles follow IBM's requirements. Because of the variety of system designs on the market, however, you should try to acquire the proper documentation from the manufacturer.

Adding Adapter Boards

Memory expansion boards typically are a last-resort way to add memory. For many systems (such as older models from Compaq) with proprietary local bus memory-expansion connectors, you must purchase all memory-expansion boards from that company. Similarly, IBM used proprietary memory connectors in the PS/2 Model 80 systems. For other industry-standard systems that use nonproprietary memory expansion (such as the IBM PC, XT, and AT) and most IBM-compatible systems, as well as most PS/2 systems, you can purchase, from hundreds of vendors, memory-expansion boards that plug into the standard bus slots.

Unfortunately, any memory expansion that plugs into a standard bus slot runs at bus speed rather than at full system speed. For this reason, most systems today provide standard SIMM-connector sockets directly on the motherboard so that the memory can be plugged directly into the system's local bus. Using memory adapter cards in these systems only slows them. Other systems use proprietary local bus connectors for memory-expansion adapters, which can cause additional problems and expense when you have to add or service memory.

In some cases, an adapter board can use slower memory than would be required on the system motherboard. (Memory adapters for PS/2 Models 50 and 60, for example, use 120-ns memory chips.) Many systems run memory-expansion slots at a fixed slower speed—8 MHz for most ISA bus systems—so that installed adapters function properly. The PS/2 system memory adapters may be able to run more slowly than main memory because of the Micro Channel Architecture (MCA) interface's higher level of controls and capabilities. The MCA's asynchronous design enables adapters to remain independent of the processor's speed and to request additional wait states, as required, to accommodate the slower adapters.

Installing Memory

This section discusses installing memory chips—specifically, new RAM chips or memory modules. The section also covers the problems that you are most likely to encounter and how to avoid them; it also provides information on configuring your system to use new memory.

When you install or remove memory, you are most likely to encounter the following problems:

- Electrostatic discharge
- Broken or bent pins
- Incorrect switch and jumper settings

To prevent electrostatic discharge (ESD) when you install sensitive memory chips or boards, do not wear synthetic-fiber clothing or leather-soled shoes. Remove any static charge that you are carrying by touching the system chassis before you begin, or—better yet—wear a good commercial grounding strap on your wrist. You can order one from an electronics parts store or mail-order house. A grounding strap consists of a conductive wristband grounded at the other end by a wire clipped to the system chassis (usually with an alligator clip). Leave the system unit plugged in—but turned off—to keep it grounded.

Caution

Be sure to use a properly designed commercial grounding strap; *do not make one yourself*. Commercial units have a one-megohm resistor that serves as protection if you accidentally touch live power. The resistor ensures that you do not become the path of least resistance to the ground and therefore become electrocuted. An improperly designed strap can cause the power to conduct through you to the ground, possibly killing you.

Broken or bent leads are another potential problem associated with installing individual memory chips (DIPs) or SIPP modules. Sometimes, the pins on new chips are bent into a *V*, making them difficult to align with the socket holes. If you notice this problem on a DIP chip, place the chip on its side on a table, and press gently to bend the pins so that they are at a 90-degree angle to the chip. For a SIPP module, you may want to use needle-nose pliers to carefully straighten the pins so that they protrude directly down from the edge of the module, with equal amounts of space between pins; then you should install the chips in the sockets one at a time.

Caution

Straightening the pins on a DIP chip or SIPP module is not difficult work, but if you are not careful, you could easily break off one of the pins, rendering the chip or memory module useless. Use great care when you straighten the bent pins on any memory chip or module. You can use chip-insertion and pin-straightening devices to ensure that the pins are straight and aligned with the socket holes; these inexpensive tools can save you a great deal of time.

Each memory chip or module must be installed to point in a certain direction. Each chip has a polarity marking on one end. This marking may be a polarity notch, a circular indentation, or both. The chip socket may have a corresponding notch. Otherwise, the motherboard may have a printed legend that indicates the orientation of the chip. If the socket is not marked, you should use other chips as a guide. The orientation of the notch indicates the location of Pin 1 on the chip. Aligning this notch correctly with the others on the board ensures that you do not install the chip backward. Gently set each chip into a socket, ensuring that every pin is properly aligned with the connector into which it fits, and push the chip in firmly with both thumbs until the chip is fully seated.

SIMM memory is oriented by a notch on one side of the module that is not present on the other side. The socket has a protrusion that must fit into this notched area on one side of the SIMM. This protrusion makes it impossible to install a SIMM backward unless you break the connector. SIPP modules, however, do not plug into a keyed socket; you have to orient them properly. The system documentation can be helpful if the motherboard has no marks to guide you. You also can use existing SIPP modules as a guide.

Before installing memory, make sure that the system power is off. Then remove the PC cover and any installed cards. SIMMs snap easily into place, but chips can be more difficult to install. A chip-installation tool is not required, but it can make inserting the chips into sockets much easier. To remove chips, use a chip extractor or small screwdriver. Never try removing a RAM chip with your fingers because you can bend the chip's pins or poke a hole in your finger with one of the pins. You remove SIMMs by releasing the locking tabs and either pulling or rolling them out of their sockets.

After adding the memory chips and putting the system back together, you may have to alter motherboard switches or jumper settings. The original PC includes two switch blocks with eight switches per block. Switch positions 1 through 4 of a PC's second switch block must be set to reflect the total installed memory. The XT has only one switch block, which is set to reflect the number of memory banks installed on the system board but not the expansion-card memory. Appendix A, "PC Technical Reference Section," provides more detailed information about the PC and XT motherboard switch settings.

IBM AT and PS/2 systems have no switches or jumpers for memory. Rather, you must run a setup program to inform the system of the total amount of memory installed. IBM-compatible AT-type systems usually have a setup program built into the system ROM BIOS, and you must run this program after installing new memory to configure the system properly.

Most memory-expansion cards also have switches or jumpers that must be set. You often must make two settings when you configure a memory card. The first setting—a starting address for the memory on the card—usually enables the memory on the card to begin using system memory addresses higher than those used by any existing memory. The second setting is the total amount of memory installed on the card.

Because of the PS/2's influence in the market, many memory boards, as well as other types of adapter cards, are made without switches. Instead, these boards have a configuration program that is used to set up the card. The configuration is stored in a special nonvolatile memory device contained on the card; after the settings are made, the card can remember the settings permanently. The Intel AboveBoard 286 and Plus versions, for example, have this switchless setup and configuration capability. Following a menu-driven configuration program is much easier than flipping tiny switches or setting jumpers located on a card. Another benefit of software configuration is that you don't even have to open the system to reconfigure a card.

After configuring your system to work properly with the additional memory, you should run a memory-diagnostics program to ensure the proper operation of the new memory. At least two and sometimes three memory-diagnostics programs are available for all systems. In order of accuracy, these programs are the following:

- POST (Power-On Self Test)
- User diagnostics disk
- Advanced diagnostics disk
- Aftermarket diagnostics software

The POST is used every time you power up the system; you can press Ctrl-A at the opening menu to access the advanced diagnostics on the reference disk.

PS/2 systems include the user diagnostics and advanced diagnostics programs on one reference disk. The disk ensures that PS/2 owners have all three memory-test programs.

Owners of standard IBM PC systems receive (in the guide-to-operations manual) a diagnostics disk that contains a good memory test. PC owners should purchase the advanced diagnostics disk as part of the hardware-maintenance service manual package. If you have purchased this package, you should use the advanced diagnostics program.

Many additional diagnostics programs are available from aftermarket utility software companies. Several companies listed in Appendix B have excellent diagnostics and test software for testing memory and other components of a system. These programs are especially useful when the manufacturer of the system does not provide its own diagnostics.

Installing 640K on an XT Motherboard

This section describes how to install 640K of RAM on the system board in an IBM XT and an IBM Portable. The upgrade essentially changes what IBM calls an XT Type 1 motherboard to a Type 2 motherboard.

The upgrade consists of installing two banks of 256K chips and two banks of 64K chips on the motherboard, and then enabling the memory chips by adding a multiplexer/decoder chip to an empty socket provided for it, as well as a jumper wire. The jumper wire enables an existing memory decoder chip (U44) to enable the additional memory. These modifications are relatively easy to perform and can be done with no soldering.

A memory-chip address is selected by two signals called *row-address select* (RAS) and *column-address select* (CAS). These signals determine where a value in a chip is located. You modify the signals by installing the jumper as indicated in this section so that the first two banks can be addressed four times deeper than they originally were, thus using the additional address locations in the 256K chips rather than in the original 64K chips.

To install 640K on an IBM XT motherboard, you must obtain the following parts from a chip vendor or electronics supply store (see Appendix B for sources):

- Eighteen 256K-by-1-bit, 200-ns (or faster) memory chips
- One 74LS158 (multiplexer/decoder) chip
- A small length of thin (30-gauge) jumper or wire-wrapping wire

After you have these parts, follow these steps:

1. Remove the motherboard (as explained in Chapter 4, "Motherboards").
2. Plug the 74LS158 chip into the socket labeled U84.

   All motherboard components are identified by an alphanumeric value. The letter usually indicates the type of component, and the number indicates a sequential component ID number for that type of component. The coding can differ among manufacturers, but most use this lettering scheme:

U	Integrated circuit
Q	Transistor
C	Capacitor
R	Resistor
T	Transformer
L	Coil
Y	Crystal
D	Diode

   Usually, component numbering follows a pattern in which the numbers increase as you move from left to right along the first row of components. Then the numbers begin again, at the left side and one row lower. You should be able to locate all the IC chips, starting with U1 in the upper-left corner of the board and ending with U90 in the lower-right corner.

3. Remove the IC installed in the socket labeled U44.
4. Install a jumper wire connecting Pin 1 to Pin 8 on the chip removed from position U44. To avoid making changes that you cannot undo later, it is easiest to wrap the ends of the jumper wire around the indicated IC pins. Be sure that the ends of the wire are wrapped securely around pins 1 and 8 and that the connection is good.

Route the wire on the underside of the IC so that the wire is held in place when you install the chip. Reinstall the chip with the jumper in place. The IC might sit slightly higher in the socket because of the wire underneath it, so make sure that it is seated as far as it goes.

Caution

Be careful with the U44: it is a unique IBM programmable array logic (PAL) chip that you cannot purchase separately. A PAL chip is burned with a circuit pattern, much as a ROM chip is burned with data values. The only way to obtain the chip is to purchase a new motherboard from IBM. Some chip houses, such as Microprocessors Unlimited (see Appendix B,the "Vendor List"), sell a copy of this PAL chip with the jumper wire modification burned in. Using one of these modified PAL chips is an alternative to adding the jumper wire manually, but be aware that most companies charge $20 for the chip. Adding the wire yourself is much cheaper.

Another alternative to adding the jumper wire to the chip in U44 is soldering two jumper pins into the plated holes numbered 1 and 2 in the jumper pad labeled E2. Using a standard plug-in jumper, you install a jumper across the two added pins. I normally do not recommend this method unless you are experienced with a soldering iron. All IBM XT Type 2 motherboards already have this modification built in.

5. Remove the 18 existing 64K-by-1-bit chips from banks 0 and 1. Reinstall the chips in banks 2 and 3 (if 64K chips are not already in these banks) or store them as spare chips.
6. Install the 18 new 256K-by-1-bit chips in banks 0 and 1.
7. Be sure that switches 3 and 4 of switch block SW1 are set to Off.
8. Replace the motherboard, and restore all other system components except memory cards.

   Remember that the motherboard now has 640K on it and that no other boards must address that space. If other memory cards were previously configured to supply memory in the first 640K area, you must reconfigure these cards or remove them from the system.
9. Power up the system, and test the installed memory for 640K without other memory-adapter cards.

If all operations are successful, you have performed an inexpensive memory upgrade that matches the capabilities of the Type 2 XT motherboard from IBM.

Upgrading the ROM BIOS

In this section, you learn that ROM BIOS upgrades can improve a system in many ways. You also learn that the upgrades can be difficult and may require much more than plugging in a generic set of ROM chips.

The *ROM BIOS*, or read-only memory basic input-output system, provides the crude brains that get your computer's components working together. A simple $30-to-$90 BIOS replacement can give your computer faster performance and more features.

The BIOS is the reason why DOS can operate on virtually any IBM-compatible system despite hardware differences. Because the BIOS communicates with the hardware, the BIOS must be specific to the hardware and match it completely. Instead of creating their own BIOSes, many computer makers buy a BIOS from specialists such as American Megatrends Inc. (AMI), Award Software, Microid Research, and Phoenix Technologies Ltd. A hardware manufacturer that wants to license a BIOS must undergo a lengthy process of working with the BIOS company to tailor the BIOS code to the hardware. This process is what makes upgrading a BIOS so difficult. The BIOS usually resides on ROM chips on the motherboard; some newer PS/2 systems from IBM, however, have a disk-loaded BIOS that is much easier to upgrade.

The BIOS is a collection of small computer programs embedded in an *EPROM* (erasable programmable read-only memory) chip or chips, depending on the design of your computer. That collection of programs is the first thing loaded when you start your computer, even before the operating system. Simply put, the BIOS has three main functions:

- Tests your computer's components when it is turned on. This test is called the Power-On Self-Test, or POST. The POST tests your computer's memory, motherboard, video adapter, disk controller, keyboard, and other crucial components.
- Finds the operating system and loads, or boots, it. This operation is called the bootstrap loader routine. If an operating system is found, it is loaded and given control of your computer.
- After an operating system is loaded, the BIOS works with your processor to give software programs easy access to your computer's specific features. For example, the BIOS tells your computer how to work with your video card and hard disk when a software program requires these devices.

In older systems, you often must upgrade the BIOS to take advantage of some other upgrade. To install some of the newer IDE (Integrated Drive Electronics) hard drives and 1.44M or 2.88M floppy drives in older machines, for example, you may need a BIOS upgrade. Machines still are being sold with older BIOSes that do not support the user-definable drive-type feature required for easy installation of an IDE drive or that may have timing problems associated with IDE drives.

The following list shows the primary functions of a ROM BIOS upgrade:

- Adding 720K, 1.44M, or 2.88M 3 1/2-inch floppy drive support to a system
- Eliminating controller- or device-driver-based hard disk parameter translation for MFM, RLL, IDE, or ESDI drives with 1,024 or fewer cylinders by using a user-definable hard drive type matched to the drive
- Adding support for block-mode Programmed I/O (PIO) transfers for an ATA-IDE (AT Attachment-Integrated Drive Electronics) hard disk
- Adding 101-key Enhanced Keyboard support
- Adding compatibility for Novell networks
- Adding compatibility for VGA displays
- Adding support for additional serial (COM) ports and printer ports
- Adding password protection
- Adding virus protection
- Adding support for additional floppy disk drives
- Correcting known bugs or compatibility problems with certain hardware and software

Because of the variety of motherboard designs on the market, ordering a BIOS upgrade often is more difficult than it sounds initially. If you have a name-brand system with a well-known design, the process can be simple. For many lesser-known compatible systems, however, you must give the BIOS vendor information about the system, such as the type of manufacturer's chipset that the motherboard uses.

For most BIOS upgrades, you must obtain the following information:

- The make and model of the system unit
- The type of CPU (for example, 286, 386DX, 386SX, 486DX, 486SX, and so on)
- The make and version of the existing BIOS
- The part numbers of the existing ROM chips (you may have to peel back the labels to read this information)
- The make, model, or part numbers of integrated motherboard chipsets, if used (for example, Chips & Technologies, SUNTAC, VLSI, OPTI, and others)

An *integrated chipset* is a group of chips on the original AT motherboard that can perform the functions of hundreds of discrete chips. Many chipsets offer customizable features that are available only if you have the correct BIOS. Most differences among systems today lie in the variety of integrated chipsets that are now used to manufacture PCs and in the special initialization required to operate these chips. Even IBM uses third-party integrated chipsets from VLSI in PS/2 ISA-bus systems such as Models 30-286, 35 SX, and 40 SX.

The BIOS also must support variations in keyboard-controller programming and in the way that nonstandard features such as speed switching are handled. A computer that uses the Chips & Technologies NEAT chipset, for example, must have a BIOS specifically made for it. The BIOS must initialize the NEAT chipset registers properly; otherwise, the machine does not even boot. The BIOS also must have support for this chipset's special features. Each of the 20 or more popular chipsets for 286, 386, 486, and 586 machines requires specific BIOS support for proper operation. A generic BIOS may boot some systems, but certain features, such as shifting to and from protected mode and speed switching, may not be possible without the correct BIOS.

Keyboard-Controller Chips

Besides the main system ROM, AT-class computers also have a keyboard controller or keyboard ROM, which is a keyboard-controller microprocessor with its own built-in ROM. The keyboard controller usually is an Intel 8042 microcontroller, which incorporates a microprocessor, RAM, ROM, and I/O ports. The keyboard controller usually is a 40-pin chip, often with a label that has a copyright notice identifying the BIOS code programmed into the chip.

The keyboard controller controls the reset and A20 lines and also deciphers the keyboard scan codes. The A20 line is used in extended memory and other protected-mode operations. In many systems, one of the unused ports is used to select the CPU clock speed. Because of the tie-in with the keyboard controller and protected-mode operation, many problems with keyboard controllers become evident when you use either Windows or OS/2. If you experience lockups or keyboard problems with either Windows or OS/2 software—or with any software that runs in protected mode, such as Lotus 1-2-3 Release 3.x—get a replacement from your BIOS vendor or system-board vendor.

IBM systems do not need a replacement of the keyboard controller for upgrade purposes. (Replacement is difficult because the chip normally is soldered in.) Most manufacturers of IBM-compatible systems install the keyboard-controller chip in a socket so that you can upgrade or replace it easily. If you upgrade the BIOS in your system, the BIOS vendor often includes a compatible keyboard controller as well. You usually do not have to buy the controller unless your old keyboard controller has a problem with the new BIOS.

BIOS Manufacturers and Vendors

Several BIOS manufacturers have developed ROM BIOS software for use in upgrading IBM or IBM-compatible systems. The following companies are the largest manufacturers of ROM BIOS software:

- Phoenix
- American Megatrends International (AMI)
- Microid Research (MR)
- Award

Phoenix pioneered the IBM-compatible BIOS and the legal means to develop a product that is fully compatible with IBM's BIOS without infringing on the corporation's copyright. Phoenix first introduced many new features, such as user-defined hard drive types and 1.44M drive support. The Phoenix BIOS has a very good Power-On Self Test; this thorough POST presents a complete set of failure codes for diagnosing problems, especially the ones that occur when a system seems to be dead. (Appendix A contains a complete list of Phoenix BIOS POST error codes.)

The Phoenix BIOS documentation, a complete three-volume reference package, is one of the product's most useful features. This documentation includes *System BIOS for IBM PC/XT/AT Computers and Compatibles*, *CBIOS for IBM PS/2 Computers and Compatibles*, and *ABIOS for IBM PS/2 Computers and Compatibles*. I recommend these excellent reference works, published by Addison-Wesley, even if you do not have the Phoenix BIOS (although some of its specific information does not apply to other systems).

The BIOS produced by AMI is very popular and surpasses even Phoenix in new system installations. The AMI BIOS offers a less comprehensive Power-On Self Test than the Phoenix BIOS does, but it has an extensive diagnostics program in ROM. You even can purchase the program separately, as AMIDIAG. The in-ROM version, however, lacks the capability to test memory—a crucial capability if the failure is in the first bank. On the other hand, the BIOS is very compatible with the PC standard, available for several different chipsets and motherboards, and has been handled responsibly from the support level. When problems have occurred, AMI has fixed them, earning this program full compatibility with OS/2 and other difficult environments.

Because AMI manufactures its own motherboards, it has a distinct advantage over other BIOS companies. If the motherboard and BIOS are made by the same source, the single vendor probably can resolve any interaction problems between the BIOS and motherboard quickly, without shifting blame for the problem to another party. I recommend buying AMI's motherboards, because you generally don't have to worry about compatibility problems between the AMI BIOS and the AMI motherboard. Even if problems occur, AMI corrects them.

Microid Research is newer than some of the other BIOS manufacturers, but its BIOS has proved to be very compatible. The MR BIOS, as it is called, supports several CPU and motherboard chipset combinations. The MR BIOS offers one of the easiest and most informative setup programs available; this program does a good job of explaining the options. Some of you know that setting the advanced chipset functions is not always intuitive with the other BIOS vendors' products. In short, I recommend that you look at the MR BIOS if you are considering a BIOS upgrade or configuring a bare motherboard from scratch.

Award, the third-largest manufacturer of BIOS software, has made a name for itself with many system vendors, because it licenses the BIOS code to them for further modification. AST, for example, purchased the rights to the Award BIOS for its own systems and now can modify the BIOS internally, as though it created the BIOS from scratch. In a sense, AST could develop its own custom BIOS, using the Award code as a starting point.

Award also provides precustomized BIOS code for manufacturers. Although Award's BIOS is not yet as popular as the Phoenix and AMI BIOSes, it is very popular, and compatibility, even in tough environments such as OS/2, is ensured.

If you want to replace or upgrade your BIOS, you can obtain replacement chips directly from the BIOS manufacturer or from the following recommended distributors:

- *Micro Firmware Inc.* Micro Firmware offers an extensive line of Phoenix BIOS upgrades, with more than 50 common 8088, 286, 386, and 486 versions available. This company develops BIOS upgrades for specific hardware platforms, even when the original motherboard manufacturer is no longer in business. Many other BIOS vendors sell BIOSes developed by Micro Firmware for specific platforms.

- *Washburn & Company Distributors.* This licensed AMI distributor deals exclusively with AMI BIOS upgrades. Washburn has complete AMI motherboard and BIOS packages. A primary distributor for AMI, Washburn has great expertise in dealing with BIOS upgrade problems. The company also sells Second Nature, a disk drive support product that may eliminate the need for a BIOS if all you want is additional hard disk or floppy drive support.

- *Microid Research.* This company manufactures and sells its own MR BIOS direct, and offers technical support as well.

Alternatives to Upgrading the BIOS

Some companies, especially those that purchase only IBM systems, are timid about removing the BIOS and replacing it with a BIOS made by a company other than IBM. If you are leery of removing your system BIOS, another solution is available: you can purchase an accessory ROM to augment, rather than replace, your existing BIOS.

Second Nature is one of the best of these products. An accessory ROM for IBM and compatibles, Second Nature adds the following important features to a system:

- A greatly expanded BIOS hard disk parameter table

- Support for 720K and 1.44M drives

No software drivers or other memory-resident programs are used when Second Nature is installed. The hard disk parameters supplied by Second Nature overlay existing drive parameters in your system ROM BIOS. Second Nature is compatible with all popular operating systems (including DOS, Windows, and OS/2) and also has a built-in, low-level, format-and-verify utility.

Most AT-type computers have four 28-pin ROM sockets, two of which normally are occupied by the existing ROM BIOS chips. If no additional empty sockets are available (such as in the IBM XT-286 or some early IBM ATs), or if your computer cannot load accessory ROMs in the E000 memory segment (as in Sperry, Leading Edge, and some other AT clones), you can buy an optional card to install the Second Nature ROMs in any 8- or 16-bit slot. For more information about this product, contact Washburn & Company Distributors (see Appendix B).

Special ROM BIOS-Related Problems

Some known problems exist in certain ROM BIOS versions as well as in some systems sold during the past few years. Several of these problems have the potential to affect a large number of people, either because the problem is severe or many systems have the problem. This section describes some of the most important known BIOS- and system-interaction problems and also provides solutions for the problems.

Some systems with BIOSes even as recent as 1992 or 1993 may not start after you upgrade to DOS 6.x. Some of the older BIOSes came on line when DOS 3.3 or earlier was the current operating system. As a result, those older BIOSes often cannot take advantage of the advanced features of DOS 6.x.

If you use an AT&T 6300 system, you will want to use BIOS Version 1.43, which is the most recent one made for the system. This version solves many problems with older 6300 systems and also provides support for a 720K floppy disk drive. You can order BIOS Version 1.43 for about $35 from the AT&T National Parts Sales Center (see Appendix B) under part number 105203780.

Some systems with the AMI BIOS have had problems with IDE hard disk drives. IDE (Integrated Drive Electronics) drives have been touted as being fully port-compatible with existing ST-506/412 (MFM or RLL) and ESDI drives. Some IDE drives, however, take somewhat longer than they should after certain commands to present valid data at their ports. In late 1989, AMI received many reports of problems with IDE drives, especially Conner and Toshiba drives. Because of these timing problems, AMI BIOS versions dated earlier than April 9, 1990, are not recommended for use with IDE drives, and data loss can result if earlier versions are used. You may experience `Drive C not ready` errors with certain IDE drives, such as those from Conner Peripherals. If you have a computer with an IDE drive and an AMI BIOS dated earlier than 04/09/90, you should get a newer BIOS from the system vendor.

To make sure that you have the correct AMI BIOS version, look for this figure in the lower-left corner of the screen when you boot your computer:

```
xxxx-zzzz-040990-Kr
```

The `040990` indicates a BIOS date of April 9, 1990—the minimum version to use. Older versions are OK only if you are *not* using IDE drives. The `xxxx-zzzz` indicates the BIOS type code and an OEM (original equipment manufacturer) ID number. For AMI-manufactured motherboards, for example, the BIOS type code is DAMI-[model code]. The `r` indicates the keyboard-controller chip's revision level, which should be revision F or later to avert problems.

Changes to an Existing BIOS

If you have access to the correct tools and knowledge, you can perform some interesting modifications or upgrades to your system by altering your existing ROM BIOS. This section discusses several modifications that I have performed on my own systems. (These modifications have worked *for me*, and I am not necessarily recommending that anyone else perform them.) If nothing else, the research and development of these modifications taught me much about the way some things work in an IBM-compatible system, and I know that many of you are interested in some of this information.

Note

These types of modifications are for readers who are especially technically astute or extremely adventurous; they are not recommended for everyone, especially for readers for whom system reliability is of crucial importance. But the following information should prove to be interesting even for readers of this book who do not attempt these operations on their systems.

Using EPROM Programming Equipment. You can accomplish some interesting modifications or upgrades and even repairs of a system by using an EPROM programmer (or *burner*, as it is sometimes called). *EPROM* stands for erasable programmable read-only memory—a type of chip that can have a program burned, or fused, into it by way of an EPROM programmer device. These devices cost anywhere from about $100 to several thousands of dollars, depending on the capabilities of the device.

Most cheaper EPROM programmers are more than adequate for burning PC ROMs. Both JDR Microdevices and Jameco Electronics sell EPROM programmers; I use and recommend one made by Andromeda Research (see Appendix B). These programmers either connect to a slot or use the standard serial or parallel ports for communications. The Andromeda Research EPROM burner connects to a system parallel port that gives it flexibility and performance that are not equaled in most other units. To erase an EPROM, you also need an EPROM eraser, an inexpensive device ($30 to $100) that exposes the EPROM chip to intense ultraviolet light for about three to five minutes. I recommend and use a simple unit made by DataRace (see Appendix B).

With an EPROM programmer, you can modify or customize your system ROM BIOS, as well as the ROM chips found on many expansion cards. You also can add hard drives to the drive table, change sign-on or error messages, and make specific changes to increase performance or otherwise customize your system. The ability to alter ROMs gives you an extra level of capability in upgrading and repairing systems.

Note

You also can use an EPROM programmer to work with ROMs in other types of computer systems. I have even altered the ROMs that contain the programs and data tables for the electronic control module (ECM) in several General Motors automobiles, giving me total control of turbocharger boost settings, engine-temperature and electric-fan calibrations, vehicle speed governors, torque converter clutch operation, and even fuel-injector and spark-advance curves.

Backing up Your BIOS. One often-overlooked benefit of an EPROM programmer is that you can use it to essentially back up your ROMs in case they are later damaged. Many hardware vendors, such as IBM, do not offer ROM upgrades for their systems, and the only way to repair a motherboard or card with a damaged ROM is to burn a new copy from a backup. The backup can be in the form of another EPROM chip with the original program burned in or even in the form of a disk file. I keep files containing images of the ROMs in each of my motherboards and expansion cards, in case I have to burn a new copy to repair one of my systems.

To create a disk-file copy of your motherboard ROM BIOS, you can place the ROM in an EPROM programmer and use the function provided by the device to read the EPROM into a disk file, or you can use the DOS DEBUG program to read your ROM BIOS from memory and transfer it to disk as a file. To use DEBUG in this manner, follow these instructions:

```
C:\>DEBUG                      ;Run DEBUG
-N SEG-F.ROM                   ;Name the file
-R BX                          ;Change BX register (high-order file size)
BX 0000                        ; from 0
:1                             ;  to 1 (indicates 64K file)
-M F000:0 FFFF CS:0            ;Move BIOS data to current code segment
-W 0                           ;Write file from offset 0 in code segment
Writing 10000 bytes            ;  10000h = 64K
-Q                             ;Quit DEBUG
```

These instructions save the entire 64K segment range from F000:0000 to 000:FFFF as a file by first setting up the name and size of the file to be saved and then moving (essentially, copying) the ROM BIOS code to the current code segment when DEBUG was loaded. The data then can be written to the disk. IBM AT systems and most compatibles have only 64K of BIOS, but IBM PS/2 systems normally have a full 128K of BIOS code that resides in both segment E000 and F000. For these systems, repeat the procedure, using E000:0 as the starting address in the Move command (rather than F000:0) and, of course, a different file name in the Name command. One important quirk of this procedure is that the commands should be entered in the order indicated here. In particular, the Name command must precede the Move command, or some of the data at the beginning of the current code-segment area will be trashed.

You also can use this routine to back up any adapter-board ROMs installed in your system. To back up the ROM on the IBM SCSI adapter with cache installed in my system, for example, a similar DEBUG session works. You must know the ROM's starting address and ending address or length to continue. My SCSI adapter ROM is located at D400:0 and is 16K, or 4,000H (hex) bytes long, which means that it ends at D400:3FFF. To save this ROM as a file, execute these instructions:

```
C:\>DEBUG                      ;Run DEBUG
-N SCSI.ROM                    ;Name the file
-R CX                          ;Change CX register (low-order file size)
CX 0000                        ;  from 0
:4000                          ;  to 4000 (indicates 16K file)
-M D400:0 3FFF CS:0            ;Move BIOS data to current code segment
-W 0                           ;Write file from offset 0 in code segment
Writing 04000 bytes            ;  4000h = 16K
-Q                             ;Quit DEBUG
```

Most EPROM programmers are supplied with software that runs in your PC and enables you to control the unit. Available functions include the capability to read a ROM and save it as a disk file or write a ROM from a disk file, and the capability to copy or test ROMs. The software also should be capable of splitting a file into even and odd addresses for 16- or 32-bit systems, as well as combining two split files into one. Another requirement is the capability to calculate the proper checksum byte (usually, the last byte) in the ROM so that diagnostics passes it. All the programmers mentioned have these functions and more.

Removing the POST Speed Check. One problem in the IBM AT and XT-286 systems is that the clock speed is checked during the Power-On Self Test (POST). Checking the clock speed probably is a good idea, but it causes problems if you want to take advantage of the socketed clock crystal in these systems for a cheap and easy speedup. Most IBM compatibles do not have this speed check and can run at different clock rates, usually with no modifications of the BIOS required.

In IBM systems, the Verify Speed/Refresh Clock Rates test checks the system refresh rate (clock speed) to ensure that it is 6 or 8 MHz, depending on which IBM system you have. A marginally faster or slower rate causes the test to fail and results in a POST error of one long and one short beep, followed by a halt (`HLT`) instruction.

This test occurs at POST checkpoint 11h, which is sent to the manufacturer's test port 80h. A failure of this test, as indicated by one of the POST-card products that reads this port, can be identified by reading 11h as the last value sent to the manufacturer's test port. To eliminate the test and enable a faster-than-normal clock rate, you must patch the instruction at the proper location from a 73h (JAE—Jump if Above or Equal) to an EBh (JMP—Jump unconditionally); this patch causes the test for an abnormally high refresh rate (resulting in a low test value) to pass through the JAE instruction. When this instruction is changed to a JMP instruction, the test never passes through and falls into the error routine, no matter how fast the rate.

Notice that the test for a slow refresh rate still is intact and that it fails if the clock is below 6 or 8 MHz, depending on the system. The JAE instruction occurs at F000:05BC in IBM AT systems with the 06/10/85 or 11/15/85 ROM BIOS versions and at F000:05C0 in XT-286 systems. The original AT system with the 01/10/84 BIOS *does not have this test*. By creating a new set of chips with this value changed by an EPROM programmer, you can eliminate this speed check and enable faster clock rates.

Modifying ROM BIOS Hard Disk Drive Parameter Tables. Probably the most common change made to a BIOS is adding or changing drives in existing hard drive tables. For example, I have added two new drive types to one of my systems. Those types—25 and 26—have these parameters:

Type	Cylinders	Heads	WPC	Ctrl	LZ	S/T	Meg	MB
25	918	15	65535	08h	918	17	114.30	119.85
26	918	15	65535	08h	918	26	174.81	183.31

See Chapter 14, "Hard Disk Drives and Controllers," for more information about these parameters. In my old AT system, these table entries originally were unused (zeros), as are the remainder of types from 27 through 47. By burning a new set of ROMs with these two new completed entries, I can use my Maxtor XT-1140 drive to maximum capacity with an MFM controller (as type 25) or an RLL controller (as type 26). This setup precludes the need for a controller BIOS to override the motherboard table values and also saves me some memory in the C000 or D000 segments, where a hard disk controller ROM normally would reside. The procedure also makes my system more standard. If you are interested in performing this modification, get the *IBM AT Technical Reference Manual*, which documents the position and format of the drive tables in the BIOS.

Changing the Hard Disk Controller Head Step Rate. Another, more complicated modification that you can perform is to increase the stepping rate of the hard disk controller. The first edition of this book briefly mentioned this modification, and a reader wrote to me to express interest in it. This edition explains the precise nature of this modification and what is being changed, to give you greater insight into how the BIOS and disk controller work together. The performance gain, in fact, is relatively slight; I see making the change as being a learning experience more than anything else.

The Western Digital AT Controllers (1002/1003/1006) used by IBM in the original AT system, as well as in other compatible controllers (such as those from Data Technology Corporation and Adaptec), have a default head-stepping rate of 35μsec (microseconds). The fastest usable step rate is 16μsec—more than twice as fast. Most ST-506/412 hard disks have optimum stepping rates as low as 10μsec. By decreasing the rate to 16μsec, you can improve the seeking performance of the drive, resulting in an improvement of several milliseconds or more during an average seek. Because the standard rate is so slow compared with the optimum rate, many ST-506/412 drives—especially fast-seeking drives—do not perform to their manufacturer-rated seek performance unless the step rate from the controller is optimized in this manner.

You can decrease the step rate to 16μsec in two ways. The easiest and best way is to simply set a jumper on the controller card. Not all cards, however, support this option. In fact, only one of the many Western Digital ST-506/412 controllers—the WD1003-WAH (ST-506/412 MFM, no floppy support)—supports this option. None of the other WD cards for the AT permits you to change the step rate by way of a jumper. Adaptec, on the other hand, has a jumper for selecting the step rate on all its AT ST-506/412 controllers. Other than Adaptec's cards, most cards do not have a jumper and require other means of changing the step rate.

The second method of changing the step rate is universal and works with virtually all AT bus ST-506/412 controllers, regardless of whether they are MFM or RLL or whether they support floppy drives. You make the change by changing two bytes in the ROM BIOS hard disk support code, which alters the way that two specific commands are sent to the controller card.

First, consider a little background on the way that the BIOS and controller operate. When DOS reads or writes data to or from a hard disk, it accesses the disk through the ROM BIOS. Specifically, DOS uses the Int 13h (h = hexadecimal) functions provided in the BIOS. Int 13h functions are commands incorporated into the BIOS of an AT system that enable DOS (or any other software) to perform specific commands in relation to the drive. These Int 13h commands then are translated into direct controller-register-level commands by the BIOS. The direct controller commands are called *command control block* (CCB) commands, because each command must be presented to the controller in the form of a 7-byte command block with the command byte itself being the last (seventh) one. The other six bytes contain information such as the number of sectors on which to operate, as well as the cylinder and head positions where the command operates.

Sixteen Int 13h commands are available for hard disks in the IBM AT BIOS. Some of these commands, such as the Get Disk Type BIOS command, perform functions that do not involve accessing the controller or drive. Most other commands, however, are translated by the BIOS to the required code to send one of eight total CCB commands to the controller.

Table 17.2 shows the Int 13h BIOS commands and the specific CCB commands that the BIOS executes in the process.

Table 17.2 Int 13h AT Hard Disk BIOS and AT Controller CCB Commands

BIOS Command	Description	CCB Command	Description
00h	Reset Disk System	91h 10h	Set Parameters Recalibrate
01h	Get Status of Last Operation	—	—
02h	Read Sectors	20h	Read Sector
03h	Write Sectors	30h	Write Sector
04h	Verify Sectors	40h	Read Verify
05h	Format Track	50h	Format Track
08h	Read Drive Parameters	—	—
09h	Initialize Drive Characteristics	91h	Set Parameters
0Ah	Read Long	22h	Read Sector
0Bh	Write Long	32h	Write Sector
0Ch	Seek	70h	Seek
0Dh	Alternate Hard Disk Reset	10h	Recalibrate
10h	Test for Drive Ready	—	—
11h	Recalibrate Drive	10h	Recalibrate
14h	Controller Internal Diagnostic	80h	Diagnose
15h	Get Disk Type	—	—

Although only eight CCB commands are specific to the standard Western Digital (or compatible) AT hard disk controller, variations of some of the commands exist. Each CCB command consists of a single byte, with the most significant four bits (bits 4–7) of the byte indicating the actual command and the least significant four bits (bits 0–3) indicating various command options. For two of the CCB commands, the option bits indicate—and, in fact, set—the step rate for the controller. By changing these bits, you also change the default step rate.

Table 17.3 shows the eight standard CCB commands.

Table 17.3 WD1002/WD1003/WD1006 AT Hard Disk Controller CCB Commands

CCB Command	Description
10h-1Fh	Recalibrate
20h-23h	Read Sector
30h-33h	Write Sector
40h-41h	Read Verify
50h	Format Track
70h-7Fh	Seek
80h	Diagnose
91h	Set Parameters

In each of these commands, the CCB command byte is sent to the controller as the seventh byte of the total command block. The primary value is the first one listed in the chart. By setting the option bits, you can alter the command function. For example, the Read Sector command is 20h. By adding +1h to the command byte (making it 21h), you disable automatic retries in case of errors. This step prevents the controller from automatically rereading the sector as many as 19 additional times, in some cases; instead, the error is reported immediately. For error-correction code (ECC) errors, the controller simply attempts an immediate ECC correction, rather than rereading the sector as many as eight times in attempting to get a good read before making the ECC correction.

You can disable retries only for the Read Sector, Write Sector, and Read Verify commands. This capability is especially useful during low-level formatting, surface analyzing, or even just Read Verify testing of the drive, because any errors are reported more accurately without automatic retries taking place.

You can set another option for the Read Sector and Write Sector commands. This option involves the ECC bytes (an additional four bytes of data past the data area of the sector); it makes the sector read or write include a total 516 bytes instead of the normal 512. To set the “long” option, add +2h to the Read Sector or Write Sector CCB byte value. To change the standard Read Sector command (20h) to include the ECC bytes, for example, you add +2h, which results in a CCB command byte value of 22h. The option to include the ECC bytes during a read or write is especially useful in testing the ECC circuitry on the controller by specifically writing incorrect values and then reading them back to see the ECC in action.

For the Read Sector and Write Sector CCB commands, you can combine the disable-retries option with the long option by adding the two options (+1h and +2h = +3h), a procedure that results in a CCB command byte value of 23h (Read Sector) or 33h (Write Sector).

The last two commands—Recalibrate and Seek—have options that you can set through the option bits. For these two commands only, the option bits are used to set the stepping rate for subsequent seek commands to the drive. By adding a value from 1h through Fh to the CCB command byte for the Recalibrate or Seek commands, you can change the step rate from the default of 35µsec to something else.

The following table shows the different step rates that you can set by adding the step option to these commands.

Step Option	Step Rate (µsec)
0h	35
1h	500
2h	1,000
3h	1,500
4h	2,000
5h	2,500
6h	3,000
7h	3,500
8h	4,000
9h	4,500
Ah	5,000
Bh	5,500
Ch	6,000
Dh	6,500
Eh	3.2
Fh	16

In this case, you want to add Fh to change the step rate from the default 35µsec to 16µsec. Notice that the other possible values are much too slow (500 to 6,500µsec) and that the 3.2µsec rate is much too fast for most drives. To change the Recalibrate and Seek commands to use the 16µsec step rate, you must patch the BIOS, changing the 10h and 70h to 1Fh and 7Fh, respectively. You essentially are patching the code that is executed when an Int 13h, Function 11h (Recalibrate Drive), or Function 0Ch (Seek) BIOS command is executed. These BIOS routines contain the code that sends the 10h (Recalibrate) or 70h (Seek) CCB commands to the controllers. Because the other BIOS routines that also execute the CCB Recalibrate command do so by calling the same Int 13h Function 11h code, only one patch is required to change all occurrences of the CCB Recalibrate command.

Because these two commands can be in different positions in different BIOS, the following code shows you how to find them by using the DOS DEBUG command:

```
C:\>DEBUG                    ;Run DEBUG
-S F000:0 L 0 C6 46 FE 10    ;Search ROM for Recalibrate command
F000:30CF                    ; Found it!
```

```
-S F000:0 L 0 C6 46 FE 70    ;Search ROM for Seek command
F000:309A                    ; Found it!
-Q                           ;Quit
```

The first Search command locates the code sent to the controller for a CCB Recalibrate command, and the second Search locates the Seek command. The 10h (Recalibrate) command is at F000:30D2h in this example, because the address returned by the Search command points to the beginning of the search string, not to the end.

Remember to add 3 to each location returned for the actual location of the 10h or 70h command. The 70h byte, for example, is at F000:309Dh. These "found" locations vary from system to system; this example used an IBM AT Model 339 with an 11/15/85 BIOS. Therefore, you can change these bytes to 1Fh and 7Fh, respectively. You must have an EPROM programmer to record these changes in another set of chips.

This section examined some simple changes that you can make in a ROM BIOS with the help of an EPROM programmer. The steps involved in modifying or burning the ROMs were not described, because these instructions normally are included with the programmer device that you purchase.

Using a Flash BIOS. Flash ROM is a type of EEPROM chip that is included in several systems today. *EEPROM* (electrically erasable programmable read-only memory) is a type of ROM chip that you can erase and reprogram directly in the system without using ultraviolet light and an EPROM programmer device. Using Flash ROM enables a manufacturer to send out ROM upgrades on disk; you then can load the upgrade into the Flash ROM chip on the motherboard without removing and replacing the chip. This method saves time and money for both the system manufacturer and the end user.

Normally, the Flash ROM in a system is write-protected, and you must disable the protection before performing an update, usually by means of a jumper or switch that controls the lock on the ROM update. Without the lock, any program that knows the right instructions can rewrite the ROM in your system—not a comforting thought. Without the write protection, it is conceivable that virus programs could be written that copy themselves directly into the ROM BIOS code in your system! Fortunately, I have not seen an implementation of Flash that did not have write-protection capability.

Most manufacturers that use the Flash BIOS system notify their customers when they upgrade the BIOS for a particular system line. Usually, the cost of the upgrade is nominal; if your system is new enough, the upgrade may even be free.

Using IML System Partition BIOS. IBM uses a scheme similar to a Flash ROM called Initial Microcode Load (IML). IML is a technique in which the BIOS code is installed on the hard disk in a special hidden system partition and is loaded every time the system is powered up. Of course, the system still has a core BIOS on the motherboard, but all that BIOS does is locate and load updated BIOS code from the system partition. This technique enables IBM to distribute ROM updates on disk for installation in the system partition. The IML BIOS is loaded every time the system is reset or powered on.

Along with the system BIOS code, the system partition contains a complete copy of the Reference Diskette, which provides the option of running the setup and system-configuration software at any time during a reboot operation. This option eliminates the need to boot from the Reference Diskette to reconfigure the system and gives the impression that the entire Reference Diskette is contained in ROM.

One drawback to this technique is that the BIOS code is installed on the (SCSI) hard disk; the system cannot function properly without the properly set-up hard disk connected. You always can boot from the Reference Diskette floppy, should the hard disk fail or become disconnected, but you cannot boot from a standard floppy disk.

Upgrading Disk Drives

Interfacing a system with different types of floppy and hard disk drives sometimes is a problem because of the differences in controllers and support software. This section discusses adding floppy disk and hard disk drives to a system.

Upgrading a Floppy Disk Drive

To add floppy drives to an existing system, you must connect them to a controller board. Virtually every PC, XT, or AT comes with at least one floppy disk controller. The controllers and the system BIOS routines support different drives. Following are the floppy-drive types that are commonly installed on PCs:

- 5 1/4-inch double-density (DD) drive (360K)
- 5 1/4-inch high-density (HD) drive (1.2M)
- 3 1/2-inch double-density (DD) drive (720K)
- 3 1/2-inch high-density (HD) drive (1.44M)
- 3 1/2-inch extra-high-density (ED) drive (2.88M)

With 5 1/4-inch drives, you need to be concerned about the track-width differences. The 360K drives write wider tracks than 1.2M drives do. You must be careful when writing 360K disks in a 1.2M drive if those disks have been written on by a 360K drive because of these track-width differences. All 3 1/2-inch drives, however, write the same track width, so you need not be concerned about problems related to track width.

Because the ED or HD drives can read and write DD disks as well, a system with as few as two drives—one 5 1/4-inch HD drive and one 3 1/2-inch ED drive—can read and write any disk format. When you upgrade, therefore, you should add these higher-density drives whenever possible.

DOS versions 3.3 and later support both double-density and high-density drives. DOS versions 5.0 and later support the newer ED 3 1/2-inch drives. DOS versions earlier than 3.3 support fewer drives and probably should not be used anymore. Table 17.4 lists the floppy-disk support provided in each DOS version.

Table 17.4 DOS Version Floppy-Disk-Format Support

	DOS Version							
Supported Disk Formats	**1.0**	**1.1**	**2.0**	**3.0**	**3.2**	**3.3**	**5.0**	**6.x**
5 1/4-inch DD (160K)	×	×	×	×	×	×	×	×
5 1/4-inch DD (180K)			×	×	×	×	×	×
5 1/4-inch DD (320K)		×	×	×	×	×	×	×
5 1/4-inch DD (360K)			×	×	×	×	×	×
5 1/4-inch HD (1.2M)				×	×	×	×	×
3 1/2-inch DD (720K)					×	×	×	×
3 1/2-inch HD (1.44M)						×	×	×
3 1/2-inch ED (2.88M)							×	×

Along with DOS support, the floppy drive controller is another issue of concern. Standard controllers in PC and XT systems support only double-density drives. To save money, consider adding only the double-density versions of the 5 1/4-inch and 3 1/2-inch drives to PC- or XT-type systems because they require no special controllers, cables, or software. DOS provides everything you need in the DRIVER.SYS driver file, which you can use to support the 3 1/2-inch 720K drive in systems that do not have built-in ROM support for that drive.

The HD drives use a higher data rate (500 kHz) than the DD drives (250 kHz) do. To use these higher-density drives, you must have a controller and software capable of supporting the higher data rate. Because the newer extra-high-density (ED) drives require a 1 MHz data rate—faster than most current controllers can support—you may need a controller upgrade to run those drives.

To use 1.2M or 1.44M drives in PC or XT systems, you must purchase a controller and software that can handle the high- or extra-high-density formats. Because the AT floppy controller already supports the 500 kHz data rate required by 1.2M or 1.44M high-capacity drives, no additional controller is necessary. To upgrade your system with a newer 2.88M drive, however, you must have a new controller that supports the 1 MHz data rate.

The final issue to consider when you upgrade a floppy disk drive is BIOS support. Many older systems do not have BIOS support for the newer floppy disk drives. Therefore, even if you have the correct drive, controller, and DOS version, you still cannot use the drive correctly unless your BIOS supports it.

Table 17.5 shows the IBM BIOS versions and the floppy disk drives that they support.

Table 17.5 IBM BIOS Floppy-Drive Support

System Description	ROM BIOS Date	ID Byte	Submodel Byte	Revision	5 1/4 DD	5 1/4 HD	3 1/2 DD	3 1/2 HD	3 1/2 ED
PC	04/24/81	FF	—	—	×				
PC	10/19/81	FF	—	—	×				
PC	10/27/82	FF	—	—	×				
PC-XT	11/08/82	FE	—	—	×				
PC-XT	01/10/86	FB	00	01	×		×		
PC-XT	05/09/86	FB	00	02	×		×		
PS/2 25	06/26/87	FA	01	00	×		×		
PS/2 30	09/02/86	FA	00	00	×		×		
PS/2 30	12/12/86	FA	00	01	×		×		
PS/2 30	02/05/87	FA	00	02	×		×		
PC-AT	01/10/84	FC	—	—	×		×		
PC-AT	06/10/85	FC	00	01	×	×	×		
PC-AT	11/15/85	FC	01	00	×	×	×	×	
PC-XT 286	04/21/86	FC	02	00	×	×	×	×	
PS/1	12/01/89	FC	0B	00	×	×	×	×	
PS/2 25-286	06/28/89	FC	09	02	×	×	×		
PS/2 30-286	08/25/88	FC	09	00	×	×	×	×	
PS/2 30-286	06/28/89	FC	09	02	×	×	×	×	
PS/2 35 SX	03/15/91	F8	19	05	×	×	×	×	×
PS/2 35 SX	04/04/91	F8	19	06	×	×	×	×	×
PS/2 40 SX	03/15/91	F8	19	05	×	×	×	×	×
PS/2 40 SX	04/04/91	F8	19	06	×	×	×	×	×
PS/2 L40 SX	02/27/91	F8	23	02	×	×	×	×	
PS/2 50	02/13/87	FC	04	00	×	×	×	×	
PS/2 50	05/09/87	FC	04	01	×	×	×	×	
PS/2 50Z	01/28/88	FC	04	02	×	×	×	×	
PS/2 50Z	04/18/88	FC	04	03	×	×	×	×	
PS/2 55 SX	11/02/88	F8	0C	00	×	×	×	×	
PS/2 57 SX	07/03/91	F8	26	02	×	×	×	×	×
PS/2 60	02/13/87	FC	05	00	×		×	×	
PS/2 65 SX	02/08/90	F8	1C	00	×		×	×	
PS/2 70 386	01/29/88	F8	09	00	×	×	×	×	
PS/2 70 386	04/11/88	F8	09	02	×	×	×	×	
PS/2 70 386	12/15/89	F8	09	04	×	×	×	×	
PS/2 70 386	01/29/88	F8	04	00	×	×	×	×	
PS/2 70 386	04/11/88	F8	04	02	×	×	×	×	

(continues)

Table 17.5 Continued

System Description	ROM BIOS Date	ID Byte	Submodel Byte	Revision	5 1/4 DD	5 1/4 HD	3 1/2 DD	3 1/2 HD	3 1/2 ED
PS/2 70 386	12/15/89	F8	04	04	×	×	×	×	
PS/2 70 386	06/08/88	F8	0D	00	×	×	×	×	
PS/2 70 386	02/20/89	F8	0D	01	×	×	×	×	
PS/2 70 486	12/01/89	F8	0D	?	×	×	×	×	
PS/2 70 486	09/29/89	F8	1B	00	×	×	×	×	
PS/2 P70 386	?	F8	50	00	×	×	×	×	
PS/2 P70 386	01/18/89	F8	0B	00	×	×	×	×	
PS/2 P75 486	?	F8	52	00	×	×	×	×	×
PS/2 80 386	03/30/87	F8	00	00	×	×	×	×	
PS/2 80 386	10/07/87	F8	01	00	×	×	×	×	
PS/2 80 386	11/21/89	F8	80	01	×	×	×	×	
PS/2 90 XP 486	?	F8	2D	00	×	×	×	×	×
PS/2 90 XP 486	?	F8	2F	00	×	×			
PS/2 90 XP 486	?	F8	11	00	×	×			
PS/2 90 XP 486	?	F8	13	00	×	×			
PS/2 90 XP 486	?	F8	2B	00	×	×			
PS/2 95 XP 486	?	F8	2C	00	×	×	×		
PS/2 95 XP 486	?	F8	2E	00	×	×			
PS/2 95 XP 486	?	F8	14	00	×	×			
PS/2 95 XP 486	?	F8	16	00	×	×			
PS/2 95 XP 486	?	F8	2A	00	×	×			

? = Unavailable
The ID byte, submodel byte, and revision numbers are in hexadecimal form.

Table 17.5 shows that the original PC-AT (01/10/84) BIOS supports only the 5 1/4-inch drives and that the second revision (06/10/85) adds 720K drive support. The final AT BIOS (11/15/85) supports the 1.44M drive as well. Some of this information is undocumented by IBM; in fact, IBM states that the AT never supported the 1.44M drives. However, 1.44M drives have been installed on hundreds of AT systems with a BIOS date of 11/15/85.

If your BIOS does not support a 720K drive, DOS can add support for a 720K drive by adding a DRIVPARM command in CONFIG.SYS. You also can load DRIVER.SYS, by way of a command similar to DRIVPARM, for DOS versions that do not support the DRIVPARM command. To support a 3 1/2-inch floppy disk drive on an older system that does not have this support, you can use the following commands:

For a PC/XT-class system: DRIVPARM=/D:1 /F:2 /I

For an AT-class system: DRIVPARM=/D:1 /F:2 /I /C

The /D parameter indicates which physical drive unit is being specified; in this case, 1 equals B. The /F parameter indicates a 720K drive format. The /I parameter indicates that this drive is not supported in the BIOS. The /C parameter indicates that the drive has Disk Changeline support: the drive can alert the controller that the disk has been changed. Because the PC/XT controllers do not support a disk change, you cannot use /C with those systems.

As an alternative, especially if you are using versions of DOS 4.0 or earlier (which lack the DRIVPARM command), you can use DRIVER.SYS. The following example assumes that your DOS files are in C:\DOS.

For a PC/XT-class system: DEVICE=C:\DOS\DRIVER.SYS /D:2 /F:2

For an AT-class system: DEVICE=C:\DOS\DRIVER.SYS /D:2 /F:2 /C

Deciding whether to use DRIVPARM or DRIVER.SYS can be confusing for people who are new to personal computers. For example, the parameters for DRIVPARM and DRIVER.SYS are identical, and the commands have much the same effect. The primary difference between using DRIVPARM and using DRIVER.SYS occurs after you boot your machine.

When you use DRIVPARM, the B drive has the new, correct parameters and functions as a normal 720K drive, as though your BIOS fully supported the drive. If you place a disk in the new drive and enter **FORMAT B:**, the drive properly formats the disk as a 720K disk without additional parameters.

The procedure is not so simple if you use DRIVER.SYS. The new B drive acts as though it were a 360K drive—that is, if you execute a FORMAT B: command, DOS formats the 3 1/2-inch disk to 360K. This situation takes place because DRIVER.SYS creates a new drive letter to represent the 720K format of the new drive (one drive letter higher than your last hard disk partition letter) as a substitute for B. This new drive—which on a system with a C drive is D—operates properly as a 720K drive. When you execute the command FORMAT D:, the B drive operates in 720K mode.

To avoid this floppy-drive alphabet soup, use DRIVPARM if your DOS version supports it. The first IBM DOS version to support DRIVPARM is 5.0. MS-DOS versions (and original-equipment-manufacturer versions) as early as 3.2 support the command.

DRIVPARM and DRIVER.SYS work only for adding 720K support to a system that does not have that support for the BIOS. The DRIVPARM and DRIVER.SYS commands do not work for adding 1.44M or 2.88M support to a system that lacks that support in the BIOS, even though the DOS manual seems to indicate that they work. In these cases, you must purchase a third-party driver program or upgrade your BIOS to support the new drive. A BIOS upgrade is the recommended method of adding support for 1.44M or 2.88M drives.

PC and XT controllers can connect two floppy drives internally, through the supplied cable, and to two drives externally, through a cable that you can purchase. The AT controller supports two floppy drives and two hard disks. If you add large-capacity floppy drives to a PC or XT system, you must have a new controller card and driver programs that operate with the card to support high-density drives. Numerous vendors offer these special controllers and software drivers, as well as complete kits for installing 360K, 720K, 1.44M, and even 2.88M drives in any PC, XT, or AT system.

Adding a High-Density Floppy Drive to an XT System

Adding high-density floppy drives (1.2M or 1.44M) to an XT-class (8088) machine can be a problem. Some people incorrectly believe that a new motherboard BIOS is necessary. It's not, because the XT BIOS and system hardware does not support high-density floppy drive operation.

Special high-density floppy controllers for XTs are available. These controllers, however, include their own ROM BIOS extension that provides BIOS-level support tailored for their specific controller. The main BIOS normally has nothing to do with supporting a high-density drive on an XT.

The BIOS extension on an XT high-density controller may not work properly with some motherboard BIOS versions; ask the controller manufacturer for verification. You know that the controller BIOS is having problems when you see phantom-directory problems—that is, problems with the controller's recognizing the Disk Change signal. If you have this problem, contact the manufacturer of the controller. The manufacturer can recommend a solution, such as changing the BIOS on the controller card or installing a new motherboard BIOS. Of course, an incorrectly configured drive or bad cable also can cause these problems; be sure to check your installation thoroughly.

Adding a Hard Disk to a System

When you add hard disks to a system, purchase the entire package from one company unless you are familiar with disk drive interfaces. Because certain interfaces have emerged as industry standards, you can take the plug-and-play approach to selecting hard drives and interface cards. But for a novice to upgrading and repairing PCs, this process can be confusing. The key is that your hard drive and your disk controller or interface card must be matched perfectly so that they work together in your system.

For example, older ESDI drives and controllers often experience significant compatibility problems unless the drive and the controller card are matched. Nearly all SCSI hard drives work with any SCSI host adapter, but to get the maximum performance from the drive and adapter combination, you should ensure that they offer the same type of SCSI interface—for example, Standard SCSI, Fast SCSI, and Fast/Wide SCSI. The Fast and Fast/Wide versions sometimes are called SCSI-2, but not all drives or adapters advertised as SCSI-2 support Fast or Fast/Wide connections. Consult Chapter 14, "Hard Disk Drives and Controllers," for more information about the SCSI interface and hard disks.

An IDE drive is simple to install if it is the only drive in the system. Complications arise when you attempt to use two IDE drives on one cable or to install an IDE drive in a system that also uses some other type of drive. Because the IDE specification was relatively loose until 1990, many older IDE drives have problems sharing a cable with the newer, more standard drives.

If you are installing multiple IDE drives or installing IDE drives to coexist with other types of drives (ST-506/412, ESDI, or SCSI) in a single system, you may encounter some problems. You can solve some problems by using special IDE adapters that incorporate an on-board BIOS driver for support. Other problems, such as installing two different manufacturers' drives on a single cable, can require experimentation with master/slave jumpers and other jumper settings on the drive. In many cases, you have to contact the respective drive manufacturers, which can help you with the multiple-drive configuration.

Differences among SCSI, IDE, ESDI, and old-style ST-506/412 interfaces can complicate the process of upgrading your hard drive. To learn more about hard disk installations, be sure to familiarize yourself with the information in Chapter 14.

Speeding Up a System

This section examines ways to increase a system's speed. A common and easy way to increase a system's calculating performance is to add a math coprocessor. Products that are advertised as making your system run faster also are discussed in this section, along with indications of their relative performance gains.

Another type of performance improvement involves increasing the system clock rate, changing the processor for another type, or both. The idea of replacing the processor with a faster one can extend to replacing the entire motherboard. In this case, you do not really upgrade your system—you change it entirely. This extreme upgrade is not always recommended. This section also examines the cost benefit of these upgrades.

Math Coprocessors

The next-best thing to replacing the computer's brains is giving it a calculator. A math, or numeric, coprocessor will do just that. A *math coprocessor* is a computer chip that aids the CPU by relieving it of certain mathematical (also called *floating-point*) operations.

Not all math operations use the math coprocessor. Common operations such as addition, subtraction, multiplication, and division are performed by the processor. For sophisticated math operations—such as cosines, sines, tangents, and square roots—a math coprocessor can make the number-crunching up to 100 times faster, boosting overall performance by 200 to 1,000 percent.

Unfortunately, not all software programs can benefit from a math coprocessor; a program must be written to take advantage of this mathematical speed. Traditionally, spreadsheet, scientific, graphics, and computer-aided design (CAD) software receive a boost from a coprocessor. (For example, CAD software requires many calculations to track every dot on-screen.) Modeling and simulation software also require tremendous mathematical power. Usually, programs such as word processors do not need this power, although Microsoft Word for Windows supports a coprocessor.

About one in 10 software programs supports a math coprocessor. If your work involves software that accesses a math coprocessor, and if you do indeed use sophisticated mathematical functions, a math coprocessor can be a great addition (no pun intended). At a cost of less than $150, a math coprocessor becomes a very focused upgrade.

How do you add a math coprocessor to your computer? Most computers have a socket for a math coprocessor, usually located next to the main processor. Computers that have an 80486DX or Pentium processor do not require a math coprocessor; it's already built into the chip itself. Because of space limitations, some computers don't even have a coprocessor socket. The following table lists the coprocessors needed to assist your main processor.

Processor	Coprocessor
8086	8087
8088	8087
80286	80287
80386SX	80387SX
80386SL	80387SX
80386SLC	80387SX
80386DX	80387DX or Weitek 3167
80486SX	80487SX or Weitek 4167
80486DX	Built-in (no Intel coprocessor needed) or optional Weitek 4167
80486DX2	Built-in (no Intel coprocessor needed) or optional Weitek 4167
80486DX4	Built-in (no Intel coprocessor needed) or optional Weitek 4167
Pentium	Built-in (no Intel coprocessor needed) or optional Weitek 4167

As you can see, computers based on the 8088 and 8086 microprocessors require an 8087 coprocessor, systems based on the 80386DX require the 80387DX, and so on.

The type of coprocessor is not the only information you need to order one; you also need to know the speed needed by your computer's design. Like your processor, this speed is measured in megahertz (MHz). For example, the 8087 coprocessor originally came in three speeds: 5 MHz, 8 MHz, and 10 MHz. Each coprocessor speed is designed to match the speed of the computer. (You can now purchase an 8087XC that works at all three speeds. Be sure that the chip you purchase is designed to operate at the clock rate at which the system runs the math chip. Also, be careful about static discharges when you handle these chips; the chips are delicate and expensive. For detailed information about math coprocessors, refer to Chapter 6, "Microprocessor Types and Specifications."

As an upgrade, a math coprocessor is a very selective speed improvement. A math coprocessor works with software written specifically to recognize the coprocessor—and only in limited areas within a program. A spreadsheet program, for example, does not become faster in all mathematical operations.

There are four major manufacturers of math coprocessors. Intel Corporation, manufacturer of most microprocessors, is the leader in manufacturing math coprocessors. Other manufacturers of math coprocessors include Advanced Micro Devices (AMD), Cyrix, and Weitek Corporation.

After you choose a math coprocessor, installing it is easy. You simply buy the chip, plug it in, and flip a switch or run a setup program to recognize it.

To install a math coprocessor, follow these steps:

1. Turn off the system power.
2. Remove the system unit's cover assembly.
3. Touch the power supply to ensure that you have the same electrical potential as the system unit.
4. Remove as many plug-in adapter cards as necessary to access the coprocessor socket.
5. Locate the coprocessor socket on the motherboard. In almost all systems, the math-coprocessor socket is next to the main processor.
6. Remove the math coprocessor from its protective holder, place the coprocessor in the socket, and align Pin 1 of the processor with Pin 1 of the socket. To align the pins, look for a notch in the chip, and make sure that the notch faces in the same direction as the notches on other chips.
7. Carefully press the chip straight down, making sure that all the pins are seated in the socket. Be careful not to bend the pins.
8. If you are using a PC or XT, locate the DIP switch block of the PC on the system board, and flip switch position 2 to Off. On an AT or PS/2, run the Setup program so that the computer recognizes the addition of the coprocessor.
9. Reinstall all adapter boards that you removed.
10. Replace the system unit's cover.
11. Verify the operation of the math coprocessor by using the diagnostics or advanced diagnostics disk.

In the next few sections, we examine specific math coprocessors, from the original 8087 to the 487 chip. Each can perk up your computer's performance.

8087. The 8087 math coprocessor chip is designed for the 8088 (and 8086) CPU. Regardless of the clock speed of the 8088 chip in your system, the 8087 that you install must be rated to run at the same rate of speed as the main processor or faster; otherwise, the math coprocessor will fail. In 8088-based systems, the main CPU and the coprocessor run in synchronization, meaning that both must run at the same clock speed. In an IBM XT, for example, the 8088 runs at 4.77 MHz, and so does the 8087. Your system would work if you purchase an 8087 that is designed to run faster than 4.77 MHz, but not if you use one that is rated to run slower. To find the speed of the processor (and math coprocessor) for your system, consult your system documentation.

287. Unlike systems based on the 8088 processor chip, the 286 CPU chip and the 287 chip run asynchronously, which means that processor time is not wasted by linking the operation of the two chips for the sake of constant communication between them; instead, the two chips communicate at set time intervals. In addition, the 286 and 287 chips do not run at the same clock speed; the 80287 math chip runs at two-thirds the speed of the CPU. In a 6-MHz AT system, the math coprocessor runs at only 4 MHz.

Internally, the 287 chip is the same math chip as the 8087, meaning that the union of the 286/287 is not as effective a partnership as that of the 8088/8087. The 287 math chip, however, does effectively speed certain high-level math computations in software that is designed to take advantage of this chip. To determine the speed of the math chip supported by a particular 286-based system, consult the system documentation.

You have the additional option of adding an Intel 287XL or 287XLT math chip to these systems. The XL and XLT chips are offered in a single speed rating—12.5 MHz—but run at slower speeds on slower systems. The XL and XLT chips perform about 20 percent faster than an original 287, due to an improved die or mask design.

387. The 80387 math chip runs at the same clock speed as the main CPU, even though it runs asynchronously. The 80387 coprocessor is a high-performance math chip designed specifically to work with the 386 (unlike the 80287 coprocessor or an 8087 with different pins).

The 387 chip uses a low-power-consumption CMOS design and comes in two flavors: the 387DX, designed to work with the 386DX processor; and the 387SX, designed for the 386SX, SL, and SLC processors. Originally, Intel offered several speed versions of the 387DX coprocessor; the only version that Intel offers now is a 33-MHz version that can run at slower speeds on slower systems.

Notice that Intel improved the die design of the 33-MHz-rated 387 over all previous designs; this chip performs 10 to 20 percent faster than the slower-rated chips. This improvement was made mainly in response to competition; several other chip manufacturers began making clone 387 chips that originally outperformed Intel's. With the new mask, the Intel 387 has been brought back up to par with the clone chips, and performance differences among different manufacturers' versions of the 387 chip are minor.

487. As detailed in Chapter 6, "Microprocessor Types and Specifications," the 487SX chip is not a math coprocessor chip. Rather, the 487SX is a fully functional 486 chip, with the math coprocessor enabled, that is plugged into a vacant "overdrive" upgrade socket in what Intel calls 486SX systems. When you plug in a 487 chip, it disables the 486 chip and takes over all processing, including high-level math processing. In creating the 486SX, Intel disables the math coprocessor that is functional on a fully capable 486DX chip before releasing the chip to system manufacturers. Intel markets these chips more cheaply than it does 486DX chips, enabling system manufacturers to produce 486SX systems at roughly the same cost as 386 systems.

Main-CPU Upgrades or Replacements

You can use several methods to increase the calculating speed of a PC, including replacing the processor with a more efficient unit, replacing the processor with an In-Circuit Emulator, and replacing the entire motherboard. This section describes how to increase the calculating speed of a PC-type system, including systems based on the 8088, 286, 386, and 486 CPU chips. For detailed information on these processors, as well as on the Pentium, refer to Chapter 6.

8086 and 8088 System Processor. You have to replace your 8086 or 8088 CPU chip only if, by performing in-depth diagnostics on your system, you determine that the chip has gone bad. It is quite difficult to replace one clock speed of 8086 or 8088 CPU chip with another, along with the other components that are to be replaced, to gain what is a relatively small performance increase. This type of upgrade is not recommended.

Instead, if you use a system based on the 8086 or 8088 and want a performance increase, given the cost of new 386 and 486SX systems, you should think about buying a new system. If system replacement is not an option, the following sections describe some additional upgrade options.

NEC V20 and V30 Chip. A simple way to increase the performance of an 8088- or 8086-based system is to use the more efficient alternative processors produced by NEC: the NEC V20 and the NEC V30. These processors are plugged into the system as direct replacements for the original processors. The NEC unit that you should select varies, according to the original processor in the system. The processors are exchanged as follows:

Original Processor	Replacement Processor
Intel 8088	NEC V20
Intel 8086	NEC V30

These NEC chips were created several years ago, when Intel could not meet the demand for the production of 8086 and 8088 chips. Intel licensed other manufacturers to make these chips. NEC, which was one of the licensed manufacturers, was given the chip-mask specifications for the Intel chips and complete access to the Intel processors' source code. NEC then created a new processor that mimics the Intel units but that is more efficient in several instructions. Intel has attempted to restrain NEC from manufacturing and importing these devices, but they remain available.

One problem with the NEC processor upgrades is that they are not 100 percent compatible with Intel units. IBM PS/2 Models 25 and 30, for example, cannot use the NEC chips, and some of the special disk-duplicating programs that back up copy-protected disks do not work when the NEC chips are installed. In addition, the performance gain of upgrading to these chips is as little as 5 percent.

Because of the very small performance gain and the potential for both incompatibility and legal problems, business users probably should avoid purchasing NEC chips. You may want to consider this upgrade for a home or experimental system, however, because of its low cost. The following paragraphs provide information about the NEC processor upgrades for the 8088 chip.

The original processor in an IBM XT is a 4.77 MHz 8088. The replacement—the NEC V20—is rated to run at 8 MHz, but these chips operate properly on systems rated for the slower speed. The operating speed of the chip is controlled by the motherboard clock circuitry. The V30 chip works only in systems designed for the Intel 8086.

To install the NEC processor replacement, locate the 8088 or 8086 processor on the motherboard, remove the processor, and plug the NEC device into place. NEC chips are available for about $15 to $25, depending on the chip, the speed rating, and the retailer.

This processor upgrade is very inexpensive (about $15) and easy to install, but the speed increase is so small that it is difficult to detect. You should test-run applications and time different operations with a stopwatch to determine any time savings from using the new chip. The normal increase in performance is about 5 percent to 10 percent of the former speed. If an operation originally took 10 seconds to complete, the same operation would take 9 to 9 1/2 seconds with the new NEC chip. A 5 percent performance increase for only $15, however, is not bad.

Increasing the System Clock Rate. Another option for speeding your PC or XT system is increasing the processor's clock rate. This type of upgrade is difficult to implement for PC or XT systems, because the clock crystal cannot be easily altered or replaced. The crystal is soldered, not socketed, into place, and the crystal is multiplexed—that is, it is used for functions besides running the system clock.

The crystal in an IBM XT, for example, is a 14.31818-MHz unit whose frequency is divided by 3 by the timer circuits to generate the 4.77 MHz system clock. This 14.31818-MHz crystal frequency is on the system bus (slot) connectors at Pin B30. (This signal is one of the PC bus specifications.) Many cards, such as video boards, depend on this oscillator signal's presence for their operations. The oscillator signal is divided by 4 to obtain a 3.58-MHz signal used to drive circuits in the Color Graphics Adapter. An upgrade that attempts to increase the clock rate of the main processor, therefore, probably would interfere with the original crystal and its subset function of generating a 14.31818-MHz oscillator signal for the system bus.

Several years ago, some devices sold for PC and XT systems could increase the clock rate for the main processor without interfering with other circuits on the motherboard. These devices have been unavailable for some time. A much better alternative is installing an in-circuit emulator (ICE) board or a processor-enhancement card that has a 286 or 386 processor on board, or replacing the entire motherboard.

In-Circuit Emulator (ICE) Boards. You can upgrade your PC/XT system with an In-Circuit Emulator (ICE) board that serves as an entire replacement for the 8086 or 8088 CPU chip. An ICE board plugs into one of the system's expansion slots and has a cable that plugs into the microprocessor socket from which the CPU chip has been removed. The ICE board has a 286 or 386 processor running at a high clock rate, as well as high-speed 16- or 32-bit memory on board. The ICE board takes over the system as though it were a new processor.

These boards are a relatively economical system upgrade, given the fact that they provide a tremendous performance boost.

Another feature of some of the better ICE boards is that you can install your original 8086 or 8088 processor on the ICE board. For compatibility and reliability testing, you then can use the original processor to run the system.

To install an ICE board, follow these steps:

1. Remove the main processor from the motherboard. In some units, you install the processor on the speedup board. Other units require that you store the original processor for safekeeping.
2. Plug the speedup board into an open slot—preferably, next to the processor socket.
3. Run a cable from the board to the processor socket, and plug in the cable. The installation is complete.

These units have their own on-board memory, a full complement of 640K, or a small amount of cache memory. The high-speed memory is directly accessible by the 80286 or 80386 processor on the speedup board. Your system memory still is used in some of these units.

Most of these boards do not convert a PC or XT to an AT-type system with the capability to address 16M of memory and to run OS/2 or perform other AT functions, but they give a PC or XT system the speed of an AT system.

Replacing the PC/XT Motherboard. Another way to increase system speed is to replace the original PC/XT motherboard. Replacing the motherboard can convert a PC-type system to an AT design, but you must replace many other items to complete the procedure. By the time you pay for and complete this conversion, you probably could buy a complete AT-compatible system and leave your PC/XT alone. (You also could sell the PC or XT system and use the money for options on the new AT, or donate it to charity and take the tax break.)

Because of the cost, complexity, and numerous compatibility problems with other system components and accessories, you should consider replacing the motherboard in a PC- or XT-type system only in special cases—and then only if money is no object.

Complete motherboard replacements consist of placing a complete AT-type motherboard of 286-, 386-, or 486-based design in your original PC/XT case. The replacement motherboard usually has 16-bit ISA slots—or even 32-bit EISA slots.

The original AT motherboard was much larger than the PC or XT motherboards. Normally, you cannot find a full-size AT-style motherboard to fit a PC case, because the PC has openings for only five slots in the back of the case and the slot spacing is different from that of an XT or AT system. Replacement AT-type motherboards that fit an XT case usually are called "baby AT" designs.

Installing a replacement motherboard is relatively difficult, but only because you must disassemble the entire system to replace the motherboard and then reassemble the system. When you undertake this kind of task, it is important that you mark all data and control cables to remind yourself which cables work with which device. It also is a good idea to mark the power connectors, particularly those that plug into the motherboard itself (usually marked P8 and P9), to help you remember the orientation of these connectors. If you install the motherboard power connectors backward, you may ruin the new motherboard.

The idea of replacing the motherboard sounds attractive at first, especially when you see the low prices of some of the replacement motherboards. But problems occur when you reassemble the rest of your old system. For example, are you going to use that XT hard disk controller in what now is an AT system? If you do, you get XT hard drive performance on a fast AT system. Following are some additional issues and problems that you may encounter when you create a hybrid system that is part AT and part XT:

- You must tell the Setup program that no hard disks are installed so that your XT controller's built-in ROM BIOS can operate the card properly.
- The XT controller uses Interrupt 5, which conflicts with a second parallel port (LPT2) in the AT.
- Your XT floppy controller does not support the high-density drives required to boot and run OS/2 and other software.
- For proper AT and OS/2 operation, you must change all your serial ports to those that have a 16450 or 16550A UART chip.
- Any memory that you already have in the XT system—including both chips and complete memory cards—probably does not work properly in an AT.

Replacing the motherboard has other disadvantages over the ICE-board upgrade discussed in preceding sections. The biggest disadvantage is that you have many more items to replace or upgrade after you replace the motherboard. Many peripherals and components of XT systems that continue to work after you install an ICE board are not suitable for an AT-type system.

Another disadvantage is that you are losing your original system. If you encounter a compatibility problem that prompts you to switch back to your old system configuration, you have to remove the new motherboard and reinstall the old. No switch can enable you to use your old processor until you work out conflicts or other compatibility problems, as you can in some ICE-board designs.

286 Systems. Upgrading AT-class systems is much simpler than upgrading PC- or XT-type systems, because you have fewer alternatives to consider. Also, the motherboard-replacement upgrade in an AT system works well because you start and end with an AT-class system. You do not have to change other system components, such as disk controllers, drives, serial ports, memory, and keyboards.

AT-type systems have primarily three upgrade levels:

- Increase the clock rate of the CPU by replacing the crystal oscillator with a faster one.
- Replace the existing processor with an In-Circuit Emulator (ICE) board.
- Replace the motherboard with one that contains a more powerful processor.

These upgrades are described in the following sections.

Increasing an AT System's Clock Rate. IBM's decision to socket the clock crystal in the PC AT met with much interest and speculation when the machine first appeared. The ability to unplug and replace the crystal is the key factor in attempting to increase an AT's clock rate. Increasing the speed of the system beyond its original design can result in a somewhat unstable system; a lot depends on how far you want to push.

Increasing system speed from 6 MHz to 8 MHz, or from 8 MHz to 10 MHz, is relatively conservative and usually enables the system to operate without problems. Attempting to increase the speed beyond these levels, however, can result in erratic or flaky system operation. The type of upgrade described in this section is what I call a hacker upgrade—suitable for people who want to experiment with their systems, but probably unsuitable for people who use their systems in a business environment.

Because the crystal is not multiplexed, as it is in PC and XT systems (it does not perform functions other than regulating the system clock speed), replacing it does not disturb other areas of the system. The oscillator signal required for Pin B30 in the expansion-bus connectors is generated independently by a 14.31818-MHz crystal that is separate from the main clock crystal on the motherboard. To minimize radio interference, the bus oscillator crystal is under a large, square piece of metal shielding. Because the two oscillators are separate, you can replace the processor clock crystal with a faster one and not interfere with the rest of the system. Upgrading an AT system's clock crystal is inexpensive because you don't have to supply any additional clock circuitry.

The main system clock crystal is in a 1/2-inch-by-3/8-inch silver housing that plugs into the socket near the 80286 chip, behind the center opening for the hard disk. To remove the crystal, insert a small screwdriver between the crystal and the socket, push the crystal out of the socket, and lift the crystal up and out of the retaining clip. You can insert a faster crystal to increase the operating speed of the 80286 CPU chip driven at half the crystal's speed.

For a few dollars, you can buy a 16-MHz replacement crystal and plug it into the system in place of the original 12-MHz crystal used in the original AT Models 068 and 099. Remember that because the 80286 internally divides the system clock crystal frequency by 2, it runs at half the crystal speed. The system then runs at 8 MHz—a 30 percent improvement in processing speed. If the 16-MHz crystal works and your system runs well at 8 MHz, you can try different crystals of increasing speeds to see how far your system can go. Some systems can tolerate running at speeds as high as 10 MHz, depending on several factors. You know that you have exceeded your system's capabilities if it locks up or otherwise demonstrates erratic operation. If your system doesn't run at a certain speed, reinstall the last crystal that worked properly.

The type of crystal that IBM used in the AT has a large case and thick, gold-plated leads that may differ from what you find at your local electronic-parts store. The standard crystal that IBM used is a series-mode, AT-cut, fundamental crystal with an HC-25/U case and 0.01 percent frequency tolerance. (*AT-cut* refers to the angle at which the quartz inside the crystal is sliced and has nothing to do with AT-type computers.) Any good crystal vendor should be able to supply this type of crystal.

Installing Faster RAM in an AT. If you want to try for more speed than your system seems to allow, you can experiment further to increase the maximum running speed by replacing system memory with higher-speed RAM chips. The standard-speed chip in most AT systems is 150 ns. If you replace all the memory with 100- or 120-ns chips, you may be able to take the extra step to 10 MHz or more.

This upgrade, however, is ineffective and a waste of money. Replacing your system memory with faster RAM chips is likely to result in a faster system only if the system manufacturer installed slow RAM on a motherboard that is rated for fast RAM. For example, if a system rated for 80-ns RAM chips had 150-ns chips installed in the factory, you probably could get a significant performance boost by installing 80-ns chips. But remember that you have to replace all the RAM chips on the motherboard if you decide to try this upgrade.

Replacing an AT's CPU Chip. You can replace the processor on a slower AT with one that is rated to run at higher speeds. You can replace the 80286-6, for example, with an 80286-10 or 80286-12. This upgrade also requires the installation of a clock crystal of greater frequency than the one used by your 6-MHz 80286 chip. Don't forget that you also may have to replace the math coprocessor (if you have one) with a faster chip.

This simple crystal swap works only in the original AT models with ROM BIOS dated 01/01/84. You cannot change the crystal speed of systems with 06/10/85 or 11/15/85 ROM BIOS dates; the POST in these systems detects the crystal change and automatically fails the motherboard test. You hear the audio code for motherboard failure: one long and one short beep. Fortunately, this failure is only temporary; when you reinstall the original crystal, the system works. The failed POST results from intentional changes that IBM made in the POST routines in later ROM versions.

If you want to swap CPU chips and clock chips on systems with BIOSes dated later than 01/01/84, the simplest way around the POST speed check is replacing the IBM ROM BIOS with a BIOS from Phoenix, AMI, or Award. Alternatively, you can remove the test from the IBM BIOS by changing the ROM BIOS code and reprogramming a new chip, as described in "Removing the POST Speed Check" earlier in this chapter.

Several years ago, some companies developed variable-speed devices that throttled the system up and down when the user twisted a knob. These devices also could monitor the system-reset line to detect a reboot operation, and they automatically kicked the system into low gear until the POST was passed, thus working around the speed test. These devices, unfortunately, cost much more than a simple crystal change and have for the most part been discontinued.

386 and 486 Processor Upgrade Boards for AT Systems. Various manufacturers have produced modular-design AT-type computers that enable you to replace a small daughtercard that contains the processor. In this processor-complex design, the main processor and support chips reside on a replaceable card, with the majority of system circuits still residing on the motherboard. In effect, you have a modular motherboard design. You can replace this processor card (usually at considerable expense) with one that contains a more powerful processor. Some of these modular systems enable you to replace your 286 chip with a 386; others enable you to upgrade all the way to a 486, providing you a lengthy upgrade path.

If you have a processor-complex AT motherboard design, contact the manufacturer of the system to determine the cost of an upgrade.

In-Circuit Emulator (ICE) boards for ATs are not as diverse or complicated as the ICE boards for PC/XT type systems. AT upgrade boards normally serve as direct processor replacements: they plug into the processor socket and do not take up a slot, as ICE boards do. You install these processor upgrade boards by removing the 286 chip and installing an upgrade module in the 286 socket. Processor upgrade boards are available that use the 386SX, 386DX, 486SX, or 486DX chips running at various clock speeds. Some 386-based boards have sockets for math coprocessor chips.

Because some of these upgrade boards have had compatibility problems with the original IBM AT BIOS, most are supplied with a set of aftermarket BIOS chips as well. In many cases, these new BIOS chips also amount to an upgrade, because the new BIOS gives you the enhancements that were discussed in the previous section "Upgrading the ROM BIOS," including enhanced hard disk drive support and floppy drive support.

Several processor upgrade boards are on the market, including quality offerings from Intel, Kingston, and Sigma Data. The addresses and phone numbers of these companies appear in Appendix B.

These processor-upgrade modules are a relatively simple and effective way to double the performance of older AT computers.

Replacing the AT Motherboard. You can choose among many motherboard replacements for AT systems, most of which use 486 processors and some of which include Zero Insertion Force (ZIF) OverDrive sockets that will accept the Pentium OverDrive chip when it is released. Motherboard upgrades for an AT system do not present the same problems as upgrading an XT in this manner, because you do not have to worry about changing other system components to be compatible with the new motherboard. In general, because you are both starting and ending with an AT-class system, you do not have to worry about changing the existing disk controller, memory, serial ports, floppy controller, or floppy drives, because they already are AT devices.

Make sure that the board you purchase has the correct screw holes for mounting in the existing chassis. Most AT replacement motherboards include both AT and XT form-factor mounting holes, to enable the boards to be mounted in a variety of system chassis. Some higher-performance upgrade motherboards work only in an AT chassis, because they exploit the full AT motherboard form factor and are designed specifically for the larger AT case.

For a list of recommendations for what to look for in a motherboard, see Chapter 4 "Motherboards."

386 Systems. The modular-design AT-type computer became most popular when the 486 was introduced. Buyers of 386 machines often did not want to pay the high price of a 486 during its first months on the market, but they still wanted an upgrade path to the more powerful processor chip. As with 286-based modular systems, on a processor-complex 386 the CPU chip and some support circuitry are placed on a separate card that plugs into the motherboard.

Replacing the 386 CPU card with one equipped with a 486 chip can be quite expensive. Contact the manufacturer of the system to determine the cost of an upgrade.

Various manufacturers provide upgrades for 386-based systems. Among the most common are plug-in boards that contain a 486SX or 486DX chip. This plug-in card usually requires you to unplug the 386 from the motherboard and plug in a card that contains the new processor and support circuitry. Some upgrade cards also require you to plug an adapter board into one of your system slots with a cable running to the replacement CPU chip card. The cost of these upgrade cards varies, depending on the manufacturer.

You can, however, replace your processor with some special products. Cyrix Corporation's Cx486DRX2 is a 486 processor upgrade for 386-based machines that technically is not a 486 chip, but a clock-doubled chip with a 1K on-board cache. The upgrade kit includes a heat sink, chip-removal tool, and utility-software disk in addition to the processor itself. The documentation includes detailed diagrams showing you how to identify and replace the CPU.

Performance gains in an upgraded 33-MHz system are very impressive. The product brings 486-class performance to many 386 systems, and speed increases are noticeable in all but disk-intensive applications.

486 Systems. Upgrades for 486-based systems are so diverse that it almost takes a scorecard to keep up with them. Within the family of 486 chips, upgrades include a wide range of chips—a fact that, more than anything, reflects a desire on the part of computer buyers to remain on the cutting edge of technology in an age when the fastest computer chip changes every couple of years.

Upgrades for 486-based systems include these options:

- 486SX to 486DX (by way of the so-called 487 upgrade)
- 486SX to 486DX (through direct CPU-chip replacement)
- Slow-clock-speed 486 to faster 486 (for example, 25 MHz to 50 MHz)
- Normally clocked CPU chip to clock-doubled (DX2) chip
- Normally clocked CPU or clock-doubled CPU to clock-tripled CPU
- 486 to Pentium (by way of the Pentium OverDrive chip)

Sorting out all these chips and chip designation numbers is quite difficult for someone who has better things to do than read computer magazines all day. But the following sections provide some general guidelines on CPU upgrades for 486-based systems.

486SX to 486DX. Generally speaking, you can upgrade a system based on the 486SX chip in two ways. The most common way to upgrade a 486SX-based system is to fill the empty CPU socket known as the OverDrive socket with a 487 upgrade. As described in Chapter 7, "Memory," the 487 upgrade is a fully functional 486 chip (with math-coprocessor functions intact) that, when plugged into the OverDrive socket, disables the original SX chip.

The second route, supported on some 486SX-based systems but not all, involves removing the 486SX chip and replacing it with a 486DX chip. This option is available only if the motherboard was designed to accept not only the 486SX but also the 486DX. Upgrading these motherboards also may require you to replace the system clock chip.

For information on the compatibility of these two upgrade options with your system, contact the manufacturer.

Slow-Clock-Speed 486 to Faster 486. Some 486 motherboards enable you to replace a slower-clock-speed 486 CPU chip with a faster 486 chip. These motherboards can support an upgrade from, for example, a 25-MHz 486DX chip to a 50-MHz 486DX chip. Manufacturers often do not advertise the fact that their motherboards support this upgrade, but many computer makers use motherboards that accept various-speed 486 chips because it is less expensive to purchase a single motherboard design that supports all the chips the manufacturer plans to use than to purchase various CPU-speed-specific motherboard designs.

For information on whether your system supports a direct upgrade from a slower-clock-speed 486 CPU to a faster chip, consult your system documentation. If the motherboard manual discusses configuration for various-speed CPU chips, chances are that you can upgrade to the fastest CPU that the motherboard supports. (Remember, however, that this type of upgrade may require replacement of your system clock chip and ROM BIOS chip.)

If this kind of upgrade appeals to you, contact your motherboard manufacturer to find out what is involved in swapping CPU chips. When the time comes to purchase a new CPU chip, be sure to shop around. Mail-order chip vendors sell the same Intel CPU chips as system manufacturers, so shopping solely based on price is a good idea.

Normally Clocked CPU Chip to Clock-Doubled (DX2) Chip. The most common route to gaining greater speed and power in a 486-based system is upgrading from a normally clocked CPU chip—for example, upgrading a 33-MHz DX to a 66-MHz DX2. Intel designed the DX2 line of chips, which operate externally at the original 25-MHz or 33-MHz chip speed but that internally run at twice the speed, so that these chips would work in existing motherboard designs. The DX2 chip greatly speeds 486-based systems. The DX2 also comes in an SX model. The 486SX2 speeds up existing 486SX systems.

As mentioned in the preceding section, the motherboards of many computer systems accept various-speed 486 chips, so direct replacement of the CPU with its clock-doubled or clock-tripled version is possible. Upgrading a DX-based system to a DX2 chip, however, is not always a plug-and-play operation; replacing the existing CPU chip with its internally faster twin also can require you to replace the system clock chip and ROM BIOS chips. This replacement usually is not necessary, though, because externally, the

CPU continues to communicate with system circuitry at the speed of the original CPU chip. You cannot perform this upgrade on some motherboards that do not support it.

Many manufacturers of systems that support upgrading from a normally clocked 486 chip to a fast 486DX2 include a special CPU chip socket on the motherboard. This socket, called a Zero Insertion Force (ZIF) socket, enables you to easily remove the old CPU chip and replace it with a faster one. Some manufacturers even allow you to trade in your old 486 CPU chip for the faster one.

486SX, DX, or DX/2 to DX4. The clock-tripled 486 CPU chip, or DX4, is another attempt to squeeze greater and greater speed out of existing motherboard designs. IBM currently is marketing a 99-MHz, clock-tripled 486 CPU chip called Blue Lightning. Like the Intel DX2 chips, Blue Lightning communicates externally with system peripherals at 33 MHz. Internally, however, Blue Lightning runs at 99 MHz. IBM is marketing the chip as being one-third faster for the same price as the DX2-66.

The company also is marketing a 25-MHz/75-MHz version of the chip. IBM's chip offers twice the internal cache of Intel's 486DX2, helping make this chip efficient at high speeds.

Some industry analysts expect Blue Lightning to compete with the Pentium for use in laptop systems. Blue Lightning uses only half the power and two-thirds the voltage of a DX2, making it ideal for laptop and notebook computers. Blue Lightning also conserves battery power through a sleep mode that enables the processor and other components to suspend active operation if the computer is idle for a certain time. The chip also is used in some non-IBM systems.

Intel released the 486DX4-75 and 486DX4-100—a clock-tripled chip based on the 25-MHz and 33-MHz DX 486, respectively. Although these clock-tripled CPU chips are most likely to be used in new midrange desktop systems and in high-end notebooks and laptops, they could be used as upgrades for systems based on normally clocked and clock-doubled CPU chips and for some 486SX systems, particularly those that have ZIF sockets. Some systems would require a new clock chip and ROM BIOS for the upgrade. Using this upgrade path may not be possible on some systems because of motherboard design limitations.

Upgrading to a clock-tripled chip should be attempted only by technically aware computer users. If you want to make this kind of upgrade, consult some of the more trustworthy computer magazines before purchasing the chip. Learning the pros and cons of this upgrade will be helpful.

486 to Pentium. For some time now, the hottest marketing tool for 486-system manufacturers has been the Pentium upgrade socket: a ZIF socket specifically designed to accept the Pentium OverDrive chip, which is expected to be released by the end of 1994.

Intel already is offering OverDrive chips that double the speed of its 486SX chips and is preparing to launch clock-tripling upgrades for 486DX units. The first 486DX2 system with a Pentium OverDrive socket shipped in 1992. Intel says that much of the two-year wait was due to overheating problems and to its desire to incorporate new technology from its 90-MHz and 100-MHz Pentiums into the chip.

Nearly every computer manufacturer is marketing systems with the promise that you can buy a 486 now and upgrade to a Pentium later. This upgrade is likely to be a simple plug-and-play operation in systems that are designed for it. You should be able to flip the lever on the ZIF socket, gently slide out the old CPU chip, orient and insert the new Pentium, and then enjoy the benefits of greatly enhanced system performance.

The compatibility problems involved in this upgrade, if any, are not known at this writing. Many system manufacturers, however, are relying heavily on the Pentium OverDrive's being easier to install than any previous CPU upgrade.

Adding High-Performance Video

Adding a high-resolution video adapter and monitor is a great way to breathe some life into a system. This type of upgrade, however, can be expensive. High-resolution, color-graphics monitors cost between $300 and several thousand dollars, depending primarily on the resolution and size of the monitor.

Video adapters usually cost less than monitors—anywhere from well under $150 to $300 or more—but some of today's cutting-edge video adapters are quite costly. For detailed information on video cards and monitors, refer to Chapter 10, "Video Display Hardware and Specifications."

At the core of your decision to upgrade your monitor and display adapter is how you plan to use your system. If you primarily use DOS-based text-mode applications, for example, you probably do not need one of today's large-screen edge-to-edge monitors or the high-end video adapters that drive them. If you use a graphical user interface such as Windows or OS/2, however, you will find that a large-screen monitor and a fast video card are near-necessities. Nothing in life seems to take as long as scrolling through a lengthy document in a Windows word processor that uses a common VGA adapter; the adapter card just cannot keep up with the incredible number of pixels that it takes to repaint a Windows screen. A problem with running Windows at normal VGA resolution (640 by 480 pixels) is that you constantly have to scroll from side to side in a document to read each line. In VGA resolution, word processors can display only about four-fifths the width of a page.

The hottest video adapters for Windows and other GUIs are VESA-Standard Local Bus (VLB) and Peripheral Component Interconnect (PCI) video cards. These cards plug into a special VESA or PCI slot, respectively, in motherboards that are designed for them, and they are capable of boosting video output to an incredible degree. The Diamond Viper (available in both PCI and VLB), for example, is one of the fastest of the Windows local bus video adapters; in real-world uses, it can paint as many as 43 million pixels per second on your screen. Other companies, such as ATI Technologies, also make extremely fast VLB and PCI video adapters.

If your system doesn't have a VLB slot, you can speed Windows and other GUIs by replacing your ISA-based video card with a graphics-accelerator video card. (These video cards also are called Windows accelerator cards.) If you have a graphics coprocessor on the video card, these accelerator cards free your PC's CPU to handle other work. The Diamond Speedstar 24X is one popular graphics-accelerator card. Numerous other companies market graphics-accelerator cards that are designed to speed your video in popular GUIs.

As mentioned earlier, running a GUI in 640 by 480 (VGA) resolution can be a problem that not only makes work harder, but also makes you less productive. Scrolling from side to side in a word processing program simply to read an entire line of a document, for example, is hardly a performance enhancement; in fact, it's the kind of thing that can make the workday seem as though it will never end.

Today's 17-inch high-resolution monitors enable you to run your system in Super VGA mode (typically, 800 by 600 resolution), which enables you to view an entire line of a document without scrolling back and forth, as well as to view more lines in the page.

Modern video adapters and high-resolution monitors not only make your work go faster, but also enable you to work with graphic images that contain millions of colors. Although not everyone needs the capability to view photographic images on his monitor, many computer users do make use of this capability.

Adding a Hardware Reset Switch

A switch that applies a full reset to your system keeps power moving to the system and rescues you from a system lockup. A reset switch saves you much time, as well as some of the wear and tear on your unit from using the power switch as a reset button. IBM and most vendors of compatibles have built reset circuitry into the motherboard and added reset switches to the front of the computer case. If your machine doesn't already have a reset switch, however, the following section teaches you how to add one. (The hardest part of adding a reset switch to your system is figuring out where to mount it.)

Adding a reset button is possible on any system, including all IBM systems, because it has a power supply that provides a Power Good signal. On most IBM-compatible computers, the Power Good signal is on the connector that plugs into the rearmost power-supply connectors. In PC and XT systems, this signal traces through the motherboard to the 8284a chip at Pin 11. When the line is shorted to ground and returned to normal, the 8284a (82284, in an AT) clock-timer chip generates a reset signal on Pin 10. The reset signal is sent to the 8088 at Pin 21, and the boot process begins. In other systems that have different processors and timer chips—for example, AT and PS/2 systems—the Power Good signal also initiates a reset if the signal is grounded and returned to normal, although the wiring details vary.

In all IBM-compatible systems, when the CPU is reset, it begins to execute code at memory location F000:FFF0, which is known as the power-up reset vector. An immediate jump instruction at this location sends the CPU's instruction pointer to the start location for the particular system ROM. The system then begins the POST. The processor and DMA chips are tested first, but before the system initiates the full POST memory test, it compares the memory location 0000:0472 with the value 1234h. If these values are equal, a *warm* start is indicated, and the POST memory tests are skipped. If any other value appears, a *cold* start occurs, forcing all memory to be tested.

This procedure is the basis of an effective reset switch. By setting the flag value at memory location 0000:0472, you can have the system perform either a cold or warm start when you press a reset button. The type of reset—a hardware reset—unfreezes a locked-up machine, unlike the Ctrl-Alt-Del software reset command.

To add a reset switch, you need these parts:

- Six inches of thin (about 20-gauge) insulated wire
- A single-pole, normally open, momentary-contact push-button switch

The idea behind installing a reset switch is to run a momentary-contact switch parallel with the Power Good line and ground. To do so, follow these steps:

1. Remove from the motherboard the power-supply connector that contains the Power Good signal.

> **Note**
>
> Check your technical-reference manual to make sure that you have the right connector and can identify the signal wire that contains the Power Good signal. Sometimes, this information also is on a sticker attached to the power supply.

2. Poke the stripped end of a wire into the hole in the power-supply connector in which the Power Good signal is carried.
3. Plug the connector, with the wire inserted, back into the motherboard.
4. Run the other end of the wire under one of the screws that secures the motherboard. The screw serves as a ground.
5. Cut the wire in the middle, bare the ends, and attach the stripped wire ends to the normally open, single-pole, momentary-contact push-button switch.
6. Run the wire and the switch outside the case.

You now should have a functioning reset button. You can mount the switch to an available place on the unit (such as an empty card bracket) in which you can drill a small hole to accept the switch.

A simple button and wire are sufficient for adding a reset switch, but as a safety precaution, you can place a 1/4-watt resistor with a value between 1K ohms and 2.7K ohms inline with the wire from the Power Good line to the switch. The reason for adding the resistor is that the Power Good signal is provided by a PNP transistor inside the power supply, with its emitter connected to the +5 volt signal. Without the resistor, shorting the Power Good signal to ground for a long period can burn out the transistor.

When you press the switch, you initiate the boot sequence. The boot process that occurs (warm or cold) depends on the status of memory location 0000:0472. The value at this location is 0000h when you first power up the system and until you press Ctrl-Alt-Del for the first time. If the last boot operation was a cold boot (an initial power-on), every subsequent time that you press the reset button, a cold boot occurs. After you press Ctrl-Alt-Del once to initiate a manual warm-boot sequence, every subsequent time that you press the reset button, you initiate a warm boot that skips the memory tests.

To eliminate the need to press Ctrl-Alt-Del after you start the system every day to “set” the reset button for subsequent warm-boot operations, you can enter a program, using DEBUG, that produces a WARMSET.COM program to run in your AUTOEXEC.BAT file. This simple program quickly sets the memory flag to indicate that a warm boot should be initiated when the reset switch is pressed.

To create WARMSET.COM, make sure that you have the DEBUG program available in the path, and then enter these commands at the DOS prompt:

```
C:\>DEBUG
-N WARMSET.COM
-A 100
xxxx:0100 MOV AX,0040
xxxx:0103 MOV DS,AX
xxxx:0105 MOV WORD PTR [0072],1234
xxxx:010B INT 20
xxxx:010D
-R CX
CX 0000
:D
-W
Writing 0000D bytes
-Q
```

Unlike the Ctrl-Alt-Del combination, the hardware reset cannot be ignored by your system, no matter how locked up the system is.

The Phoenix BIOS sets the warm-boot flag during every boot sequence, regardless of whether the boot is warm or cold. Immediately after power-up, therefore, the flag is zero, which causes the standard POST test to run. Immediately after the POST, the Phoenix BIOS sets the flag for a warm boot, because many compatibles that use the Phoenix BIOS have a reset button integrated into the system. This reset button works the same way as the button that you can construct. Because of the warm-boot flag’s automatic setting, a warm boot occurs every time you press the reset button, no matter what the last boot was. If you want a cold boot to occur (including POST), you can create a COLDSET.COM program, also using DEBUG.

To create COLDSET.COM, enter these commands:

```
C:\>DEBUG
-N COLDSET.COM
-A 100
xxxx:0100 MOV AX,0040
xxxx:0103 MOV DS,AX
```

```
xxxx:0105 MOV WORD PTR [0072],0000
xxxx:010B INT 20
xxxx:010D
-R CX
CX 0000
:D
-W
Writing 0000D bytes
-Q
```

This procedure causes a reset button to initiate a cold boot with POST tests, no matter which BIOS you have.

An interesting variation on these programs is to produce two additional companion programs called WARMBOOT.COM and COLDBOOT.COM. As their names indicate, these programs go one step further than the WARMSET.COM and COLDSET.COM programs: they not only set the flag, but also cause an immediate boot.

You may wonder why you would need that operation when you can just press Ctrl-Alt-Del, or turn the power off and on, to reboot the system. The answer is that with these programs, you can initiate the boot you want from a batch file, with no operator intervention.

I use the WARMBOOT.COM program in batch files that copy new CONFIG.SYS files to my root directory and then reboot the system automatically to initiate the new configuration. For example, one batch file copies a CONFIG.SYS file, which loads local-area network (LAN) drivers, as well as a new AUTOEXEC.BAT file, to the root directory of the C drive. The AUTOEXEC.BAT file has commands that automatically log on to the network. In seconds, my network is up and running, and I am automatically logged on, with only one command. You probably can come up with other uses for these programs.

Note

If you have DOS 6.x, you can create customized boot menus that replace this whole process. Both CONFIG.SYS and AUTOEXEC.BAT can have configuration blocks that allow you to pick one of several configurations.

To create WARMBOOT.COM, enter these commands:

```
C:\>DEBUG
-N WARMBOOT.COM
-A 100
xxxx:0100 MOV AX,0040
xxxx:0103 MOV DS,AX
xxxx:0105 MOV WORD PTR [0072],1234
xxxx:010B JMP FFFF:0
xxxx:0110
-R CX
CX 0000
:10
```

```
-W
Writing 00010 bytes
-Q
```

To create COLDBOOT.COM, enter these commands:

```
C:\>DEBUG
-N COLDBOOT.COM
-A 100
xxxx:0100 MOV AX,0040
xxxx:0103 MOV DS,AX
xxxx:0105 MOV WORD PTR [0072],0000
xxxx:010B JMP FFFF:0
xxxx:0110
-R CX
CX 0000
:10
-W
Writing 00010 bytes
-Q
```

Whether or not you have a reset button, the WARMBOOT.COM and COLDBOOT.COM programs can be useful.

Upgrading the DOS Version

An upgrade that many users overlook is an upgrade to a new version of an operating system. You can complicate a DOS upgrade by using different OEM versions of DOS, but normally, you should not have to reformat your hard disk when you upgrade from one version of DOS to another.

If you are using nonstandard drivers or other non-DOS support programs, you may have to format the disk, or use Norton Utilities or some other low-level utility to modify the disk's boot sector and root directory. This modification enables the SYS command in the new version to operate properly.

You can perform an operating-system upgrade in several ways. One way is to use the automatic installation and upgrade facility built into DOS. The automatic DOS upgrade and installation utility that comes with DOS 6.x, for example, works well. If you want to upgrade on your own, follow the simple instructions in the following section to complete the upgrade in a relatively short amount of time.

To repartition your hard disk at the same time that you upgrade to a new DOS version, you must back up your entire hard disk, repartition and reformat it, and then restore all your original files. Then you can install the new version of DOS over your original (restored) version. Depending on the type of backup hardware that you use, this procedure can be somewhat time-consuming.

Upgrading DOS the Easy Way

To perform a DOS-version upgrade the easy way, follow these steps:

1. Boot the new DOS version from drive A.
2. Transfer the system files to drive C, using this command:

   SYS C:

3. Locate the DOS disk with the REPLACE program, insert the disk into drive A, and execute this command:

   COPY REPLACE.* C:

4. Use the following command to replace all DOS transient files on drive C with new versions in all subdirectories and to enable read-only files to be overwritten:

   C:REPLACE A:*.* C: /S /R

Caution

This command replaces any file on the C drive with a file of the same name on the DOS disks, no matter where on the C drive the file is located. Any files on drive C that have the same name as any of the DOS files on drive A, therefore, are overwritten by this command.

5. Change the disk in drive A to the second DOS floppy disk, and repeat the preceding step until you have inserted all the DOS floppy disks.
6. Place the DOS boot disk back in drive A.
7. Use the following command to add any new DOS transient files to the C:\DOS directory (if you have a C:\DOS directory already in place on the hard disk):

   C:REPLACE A:*.* C:\DOS /A

8. Replace the boot disk with the second DOS floppy disk, and repeat the preceding step until you have inserted all the DOS floppy disks.

When you finish, the system is capable of booting the new DOS version from the hard disk. This method ensures that all previous DOS files are overwritten by new versions, no matter which subdirectories store those files.

Note

If you use a directory other than C:\DOS to store your DOS program files, replace C:\DOS in the preceding steps with that directory.

If you have problems with the SYS command, your original installation probably has a problem, which you must repair before you can proceed. Notice that the SYS commands in DOS 4.0 and later versions are more tolerant of different configurations on the destination drive. If you have problems, I recommend that you see *Que's Guide to Data Recovery* for additional information on this subject.

Upgrading DOS the Hard Way

To take advantage of the disk-formatting capabilities of newer DOS versions, you must use the more difficult method of upgrading the DOS version. A common use for this method is upgrading from DOS 3.3 to 6.x on a hard disk larger than 32M. Because DOS 3.3 could create disk partitions that were a maximum 32M, the disk must be currently partitioned into several volumes. Because DOS 6.x permits partitions as large as 2G (gigabytes), it's a good idea to use the upgrade as an opportunity to condense the multiple partitions into one partition that matches the total capacity of the drive.

This method requires that you first back up the data on all the partitions, because the existing partitions must be destroyed and re-created as a single partition.

To perform a DOS-version upgrade by using the full backup, format, and restore method, follow these steps:

1. Boot the system from the original (older) version of DOS.
2. Execute a complete backup of all partitions, using the backup method of your choice.
3. Boot the new version of DOS.
4. Use the new DOS FDISK command to remove the existing DOS partition or partitions.
5. Use the new DOS FDISK to create the new partition or partitions.
6. Perform a high-level format, and install the new DOS system files with this command:

   ```
   FORMAT C: /S
   ```

7. Restore the previous partition backups to the new partition, but do not restore the original system files (IBMBIO.COM and IBMDOS.COM). Your restore program should enable you to selectively ignore files such as these. BACKUP and RESTORE in DOS 3.3 and later always ignore the system files.
8. Locate the new DOS disk by using the REPLACE program, insert the disk into drive A, and execute this command:

   ```
   COPY REPLACE.EXE C:
   ```

9. Use the following command to replace all DOS transient files on drive C with new versions in all subdirectories and to enable read-only files to be overwritten:

   ```
   REPLACE A:\*.* C: /S /R
   ```

Caution

This command replaces any file on the C drive with a file of the same name on the DOS disks, no matter where on C the file is located. Any files on drive C that have the same name as any of the DOS files on drive A, therefore, are overwritten by this command.

10. Place the second DOS floppy disk in drive A, and repeat the preceding step until you have inserted all DOS floppy disks.
11. Place the boot disk back in drive A.
12. Add new DOS transient files to the C:\DOS directory with this command:

    REPLACE A:*.* C:\DOS /A

13. Replace the boot disk with the second DOS floppy disk, and repeat the preceding step until you have inserted all DOS floppy disks.

When you finish, the system is capable of booting the new DOS version from the hard disk. (If you store program files in a directory other than C:\DOS, replace C:\DOS with that directory.) This method ensures that all previous DOS files are overwritten by new versions.

Backup procedures make this method very difficult unless you have a good backup system. You also must determine how to prevent your backup system from restoring the system files.

If you are using the DOS 3.3 or later BACKUP and RESTORE commands, preventing the system files from being overwritten is much simpler. Because RESTORE never can restore the files IBMBIO.COM, IBMDOS.COM, and COMMAND.COM, the chance of overwriting them is eliminated.

You must use the new DOS version's FDISK command to remove—and especially to create—the partitions to make the newer DOS version's partitioning capabilities available. One variation on this method is substituting a low-level format operation on the hard disk for step 4 of this procedure. One advantage of not repeating the low-level format is that any existing non-DOS partitions survive; in a low-level format, nothing is left on the disk. An advantage to the low-level format is that it rewrites the sectors and tracks, which might be a good idea for some older hard disks.

Summary

This chapter examined many types of system upgrades, including the upgrade of the system's most important chip: the central processing unit. The chapter also covered adding system memory, expanding hard drive storage, adding floppy drives, speeding a system, improving the video subsystem, adding a reset switch, and upgrading to a new version of DOS. You now should have some insight into ways to make your system faster and, therefore, more useful, as well as some useful recommendations on how to avoid upgrade pitfalls that other people have endured.

Upgrading a system can be a cost-effective way to keep up with the rest of the computing community and to extend the life of your system. By upgrading, however, you also can create a Frankenstein system that has compatibility problems before you realize how much money you have poured into the system. It is important to learn when to say no to more upgrades and to recognize when it's time to buy a new system. In some cases, the cost may be high; the cost is in installing new software and hardware and then getting it to work with the rest of your PC.