CHAPTER 7

The IDE Interface

An Overview of the IDE Interface

The primary interface used to connect a hard disk drive to a modern PC is typically called IDE (Integrated Drive Electronics). An interesting fact is that the true name of the interface is called ATA (AT Attachment), which refers to the fact that this interface originally was designed to connect a combined drive and controller directly to the bus of the 1984 vintage IBM AT computer, otherwise known as the ISA (Industry Standard Architecture) bus. *IDE* is a term originated by the marketing departments of some drive manufacturers to describe the drive/controller combination used in drives with the ATA interface. Integrated Drive Electronics refers to the fact that the interface electronics or controller is built into the drive and is not a separate board, as with earlier drive interfaces. Although technically the correct name for the interface is ATA, many persist in using the IDE designation today. Throughout this book, I will use both designations because they are essentially interchangeable. If you are being picky, you could say that *IDE* refers generically to any drive interface in which the controller is built into the drive, whereas *ATA* refers to the specific implementation of IDE that is used in PCs.

Today, ATA is used to connect not only hard disks, but also CD-ROM and CD-RW drives, DVD drives, high-capacity SuperDisk floppy drives, and tape drives. Even so, ATA is still thought of primarily as a hard disk interface, and it evolved directly from the separate controller and hard drive interfaces that were used prior to ATA. This chapter covers the ATA interface in detail, as well as the original interfaces from which ATA evolved. Because the ATA interface is directly integrated into virtually all motherboard chipsets, ATA is the primary storage interface used by most PCs.

Note

Even Apple has recognized the value of ATA and has incorporated it into virtually all modern Macintosh systems.

▶▶ See "LS-120 (120MB) SuperDisk Drives," p. 647.

Precursors to IDE

A variety of hard disk interfaces have been available for PC hard disks over the years. As time has passed, the number of choices has increased, and many older designs are no longer viable in newer systems.

The primary job of the hard disk controller or interface is to transmit and receive data to and from the drive. The various interface types limit how fast data can be moved from the drive to the system and offer different features as well as levels of performance. If you are putting together a system in which performance is a primary concern, you need to know how these various interfaces affect performance and what you can expect from them. Many statistics that appear in technical literature are not indicative of the real performance figures you will see in practice. I will separate the myths presented by some of these overly optimistic figures from the reality of what you will actually see.

Several types of hard disk interfaces have been used in PC systems over the years, as shown here:

Interface	When Used
ST-506/412	1978–1989
ESDI	1983–1991
SCSI	1986–Present
ATA (IDE)	1986–Present
Serial ATA	2001–Present

Of these interfaces, only ST-506/412 and ESDI are what you could call true disk-controller–to–drive interfaces, and they are obsolete. SCSI and ATA are system-level interfaces that usually incorporate a chipset-based controller interface internally. For example, most SCSI and ATA drives incorporate the same basic internal controller circuitry. The SCSI interface adds another layer of interface that attaches the controller to the system bus, whereas ATA is a more direct bus-attachment interface. Virtually all modern disk drives use either ATA or SCSI interfaces to connect to a system.

The IDE Interface

As discussed earlier, IDE (now officially referred to as ATA) is a generic term that applies to any drive with a built-in disk controller. It is an ANSI standard, or more accurately an evolving standard with various published standard versions. The term IDE can roughly apply to any disk drive with a built-in controller, whereas ATA denotes a specific interface. Referring to the original parallel version of the interface, what we call ATA originally referred to a hard disk drive that plugged directly into a version of the AT-bus, more commonly known as the 16-bit ISA bus.

ATA is a 16-bit parallel interface, meaning that 16 bits are transmitted simultaneously down the interface cable. A new interface called Serial ATA was officially introduced in late 2000 and is being adopted in systems starting in 2001. Serial ATA (SATA) sends 1 bit down the cable at a time, enabling thinner and smaller cables to be used, as well as providing higher performance due to the higher cycling speeds allowed. SATA is a completely new and updated physical interface design, while remaining compatible on the software level with parallel ATA. Throughout this book, *ATA* refers to the parallel version, whereas Serial ATA is explicitly referenced as SATA.

The first drives with integrated ISA bus (otherwise known as AT-bus) controllers were called *hardcards*; they consisted of a hard disk bolted directly to an ISA bus controller card and plugged into a slot as a single unit. Today, any drive with an integrated controller is called an IDE drive, although normally when we say *IDE*, we really mean the specific version of IDE called ATA. No matter what you call it, combining the drive and controller greatly simplifies installation because there are no separate power or signal cables from the controller to the drive. Also, when the controller and drive are assembled as a unit, the number of total components is reduced, signal paths are shorter, and the electrical connections are more noise-resistant. This results in a more reliable design than is possible when a separate controller, connected to the drive by cables, is used.

Placing the controller, including the digital-to-analog encoder/decoder (endec) on the drive offers an inherent reliability advantage over interfaces with separate controllers. Reliability is increased because the data encoding, from digital to analog, is performed directly on the drive in a tight noise-free environment. The timing-sensitive analog information does not have to travel along crude ribbon cables that are likely to pick up noise and insert propagation delays into the signals. The integrated configuration enables increases in the clock rate of the encoder and the storage density of the drive.

Integrating the controller and drive also frees the controller and drive engineers from having to adhere to the strict guidelines imposed by the earlier interface standards. Engineers can design what essentially are custom drive and controller implementations because no other controller will ever have to be connected to the drive. The resulting drive and controller combinations can offer higher performance than earlier standalone controller and drive setups. IDE drives sometimes are called drives with embedded controllers.

The ATA connector on motherboards in many systems started out as nothing more than a stripped-down ISA bus (AT-bus) slot. These connectors contain what started out as a 40-pin subset of the 98 pins that would make up a standard 16-bit ISA bus slot. Note that smaller 2 1/2-inch ATA drives use a superset 44-pin connection, which includes additional pins for power. The pins from the original ISA bus used in ATA are the only signal pins required by a standard-type AT hard disk controller. For example, because a primary AT-style disk controller uses only interrupt request (IRQ) line 14, the primary motherboard ATA IDE connector supplies only that IRQ line; no other IRQ lines are necessary. The obsolete 8-bit XT IDE motherboard connector supplied interrupt line 5 because that is what an XT controller used. Note that even if your ATA interface is integrated within the motherboard chipset South Bridge or I/O Controller Hub chip (as it would be in newer systems) and runs at higher bus speeds, the pinout and functions of the pins are still the same as the original design taken right off the ISA bus.

◀◀ See "Motherboard Interface Connectors," p. 285.

◀◀ See "The ISA Bus," p. 300.

Note

Many people who use systems with ATA connectors on the motherboard believe that a hard disk controller is built into their motherboards, but in a technical sense the controller is actually in the drive. Although the integrated ATA ports on a motherboard often are referred to as *controllers*, they are more accurately called *host adapters* (although you'll rarely hear this term). A host adapter can be thought of as a device that connects a controller to a bus.

When IDE drives are discussed, the ATA variety usually is the only kind mentioned because it is by far the most popular—and pretty much the only type in use in modern PCs. But other forms of IDE drives existed, based on other buses. For example, several PS/2 systems came with Micro-Channel (MCA)–based IDE drives that plugged directly into a Micro-Channel bus slot (through an interposer card). An 8-bit ISA form of IDE also existed but was never very popular. Most PCs use AT-Bus (16-bit)–based IDE drives, officially known as ATA drives. Many new systems are built using the Serial ATA interface as well. The ATA/IDE interfaces are by far the most popular type of drive interfaces available.

Note

IDE is a generic name that could be given to any interface in which the controller portion of the circuit is on the drive; ATA and Serial ATA refer to specific types of IDE interfaces. Because the popular form of IDE interface is ATA, the terms often are used interchangeably, even though on a technical level that is not correct. What most people call IDE is more properly called ATA.

The primary advantage of ATA drives over the older separate controller–based interfaces and newer host bus interface alternatives, such as SCSI and IEEE-1394 (iLink or FireWire), is cost. Because the separate controller or host adapter is eliminated and the cable connections are simplified, ATA drives cost much less than a standard controller and drive combination.

▶▶ See "Small Computer System Interface," p. 514.

▶▶ See "USB and IEEE-1394 (i.Link or FireWire)—Serial and Parallel Port Replacements," p. 940.

Another advantage is performance. ATA drives are some of the highest performance drives available—but they also are among the lowest performance drives. This apparent contradiction is a result of the fact that all ATA drives are different. You cannot make a blanket statement about the performance of ATA drives because each drive is unique. The high-end models, however, offer performance equal or superior to that of any other type of drive on the market for a single-user, single-tasking operating system.

IDE Origins

The earliest IDE drives were called hardcards and were nothing more than hard disks and controllers bolted directly together and plugged into a slot as a single unit. Companies such as the Plus

Development Division of Quantum took small 3 1/2-inch drives (either ST-506/412 or ESDI) and attached them directly to a standard controller. The assembly then was plugged into an ISA bus slot as though it were a normal disk controller. Unfortunately, the mounting of a heavy, vibrating hard disk in an expansion slot with nothing but a single screw to hold it in place left a lot to be desired—not to mention the possible interference with adjacent cards because many of these units were much thicker than a controller card alone.

Several companies got the idea to redesign the controller to replace the logic-board assembly on a standard hard disk and then mount it in a standard drive bay just like any other drive. Because the built-in controller in these drives still needed to plug directly into the expansion bus just like any other controller, a cable was run between the drive and one of the slots.

These connection problems were solved in various ways. Compaq was the first to incorporate a special bus adapter in its system to adapt the 98-pin AT (ISA) bus edge connector on the motherboard to a smaller 40-pin header style connector the drive would plug into. The 40-pin connectors were all that was necessary because it was known that a disk controller never would need more than 40 of the ISA bus lines.

In 1987, IBM developed its own MCA IDE drives and connected them to the bus through a bus adapter device called an *interposer card*. These bus adapters (sometimes called *paddle boards* or *angle boards*) needed only a few buffer chips and did not require any real circuitry because the drive-based controller was designed to plug directly into the bus. The paddle board nickname came from the fact that they resembled game paddle or joystick adapters, which do not have much circuitry on them. Another 8-bit variation of IDE appeared in 8-bit ISA systems, such as the PS/2 Model 30. The XT IDE interface uses a 40-pin connector similar to, but not compatible with, the 16-bit version.

IDE Bus Versions

There have been four main types of IDE interfaces based on three bus standards:

- Serial AT Attachment (SATA)
- Parallel AT Attachment (ATA) IDE (based on 16-bit ISA)
- XT IDE (based on 8-bit ISA)
- MCA IDE (based on 16-bit Micro Channel)

Of these, only the ATA versions are used today. ATA and Serial ATA have evolved with newer, faster, and more powerful versions. The improved versions of parallel ATA are referred to as ATA-2 and higher. They are also sometimes called EIDE (Enhanced IDE), Fast-ATA, Ultra-ATA, or Ultra-DMA. Even though ATA appears to have hit the end of the road with ATA-6, Serial ATA picks up where ATA leaves off and offers greater performance plus an established roadmap for future upgrades.

Note

Many people are confused about 16- versus 32-bit bus connections and 16- versus 32-bit hard drive connections. A PCI bus connection allows for a 32-bit (and possibly 64-bit in the future) connection between the bus and the IDE host interface, which is normally in the motherboard chipset South Bridge or I/O Controller Hub (ICH) chip. However, the actual parallel ATA interface between the host connector on the motherboard and the drive (or drives) itself is only a 16-bit interface. Thus, in a parallel ATA drive configuration, you are still getting only 16-bit transfers between the drive and the motherboard-based host interface. Even so, the clock speeds of the ATA interface are high enough that one or two hard drives normally cannot supply the controller enough data to saturate even a 16-bit channel. The same is true with Serial ATA, which—although it transmits only 1 bit at a time—does so at extremely high speeds.

The obsolete XT and current parallel ATA versions have standardized on 40-pin connectors and cables, but the connectors have slightly different pinouts, rendering them incompatible with one another. MCA IDE is also obsolete, uses a completely different 72-pin connector, and is designed for MCA bus systems only.

In most modern systems you will find two ATA connectors on the motherboard. If your motherboard does not have one of these connectors and you want to attach an ATA drive to your system, you can purchase an adapter card that adds an ATA interface (or two) to a system via the ISA or PCI bus slots. Some of the cards offer additional features, such as an onboard ROM BIOS or cache memory.

Because the ATA variety of IDE is the primary one in use today, that is what most of this chapter discusses.

ATA IDE

Control Data Corporation (CDC), Western Digital, and Compaq actually created what could be called the first ATA-IDE interface drive and were the first to establish the 40-pin ATA connector pinout. The first ATA IDE drives were 5 1/4-inch half-height CDC 40MB units with integrated WD controllers sold in the first Compaq 386 systems in 1986.

Eventually, the 40-pin ATA connector and drive interface design was placed before one of the ANSI standards committees that, in conjunction with drive manufacturers, ironed out some deficiencies, tied up some loose ends, and then published what was known as the CAM ATA (Common Access Method AT Attachment) interface. The CAM Committee was formed in October 1988, and the first working document of the AT Attachment interface was introduced in March 1989. Before the CAM ATA standard, many companies that followed CDC, such as Conner Peripherals, made proprietary changes to what had been done by CDC. As a result, many older ATA drives from the late 1980s are very difficult to integrate into a dual-drive setup that has newer drives. By the early 1990s, most drive manufacturers brought their drives into full compliance with the official standard, which eliminated many of these compatibility problems.

Some areas of the ATA standard have been left open for vendor-specific commands and functions. These vendor-specific commands and functions are the main reason it is so difficult to low-level format ATA drives. To work to full capability, the formatter you are using usually must know the specific vendor-unique commands for rewriting sector headers and remapping defects. Unfortunately, these and other specific drive commands differ from OEM to OEM, clouding the “standard” somewhat. Most ATA drive manufacturers have formatting software available on their Web sites.

Note

It is important to note that only the ATA IDE interface has been standardized by the industry. The XT IDE and MCA IDE never were adopted as industry-wide standards and never became very popular. These interfaces no longer are in production, and no new systems of which I am aware come with these nonstandard IDE interfaces.

ATA Standards

Today what we call the ATA interface is controlled by an independent group of representatives from major PC, drive, and component manufacturers. This group is called Technical Committee T13 and is responsible for all interface standards relating to the parallel AT Attachment storage interface. T13 is a part of the National Committee on Information Technology Standards (NCITS), which operates under rules approved by the American National Standards Institute (ANSI), a governing body that sets rules which control nonproprietary standards in the computer industry as well as many other industries. A second group called the Serial ATA Workgroup has formed to create the Serial ATA standards that will also come under ANSI control. Although these are different groups, many of the same people are in both groups. It seems as if little further development will be done on parallel ATA past the ATA-6 (ATA/100) specification. The further evolution of ATA will be in the Serial ATA form (discussed later in this chapter).

The rules these committees operate under are designed to ensure that voluntary industry standards are developed by the consensus of people and organizations in the affected industry. NCITS specifically develops Information Processing System standards, whereas ANSI approves the process under which they are developed and publishes them. Because T13 is essentially a public organization, all the working drafts, discussions, and meetings of T13 are open for all to see.

The parallel ATA interface has evolved into several successive standard versions, introduced as follows:

- ATA-1 (1986–1994)
- ATA-2 (1996; also called Fast-ATA, Fast-ATA-2, or EIDE)
- ATA-3 (1997)
- ATA-4 (1998; also called Ultra-ATA/33)
- ATA-5 (1999–present; also called Ultra-ATA/66)
- ATA-6 (2000–present; also called Ultra-ATA/100)

Each version of ATA is backward compatible with the previous versions. In other words, older ATA-1 or ATA-2 devices work fine on ATA-4, ATA-5, and ATA-6 interfaces. In cases in which the device version and interface version don't match, they work together at the capabilities of the lesser of the two. Newer versions of ATA are built on older versions and with few exceptions can be thought of as extensions of the previous versions. This means that ATA-6, for example, is generally considered equal to ATA-5 with the addition of some features.

Table 7.1 breaks down the various ATA standards. The following sections describe all the ATA versions in more detail.

Table 7.1 ATA Standards

Standard	Timeframe	PIO Modes	DMA Modes	Ultra-DMA Modes	Speed[1]	Features
ATA-1	1986–1994	0–2	0	—	8.33	Drive support up to 136.9GB BIOS issues not addressed
ATA-2	1995–1996	0–4	0–2	—	16.67	Faster PIO modes; CHS/LBA BIOS translation defined up to 8.4GB; PC Card
ATA-3	1997	0–4	0–2	—	16.67	SMART[2]; improved signal integrity; LBA support mandatory; eliminated single-word DMA modes
ATA-4	1998	0–4	0–2	0–2	33.33	Ultra-DMA modes; BIOS support up to 136.9GB
ATA-5	1999–2000	0–4	0–2	0–4	66.67	Faster UDMA modes; 80-pin cable with autodetection

Table 7.1 Continued

Standard	Timeframe	PIO Modes	DMA Modes	Ultra-DMA Modes	Speed[1]	Features
ATA-6	2001–present	0–4	0–2	0–5	100.00	100MB/sec UDMA mode; extended drive and BIOS support up to 144PB[3]

1. Speed is MB/sec

2. SMART = Self-Monitoring, Analysis, and Reporting Technology

3. PB = Petabyte; 1PB is equal to 1 quadrillion bytes

MB = Millions of bytes

GB = Billions of bytes

CHS = Cylinder head sector

LBA = Logical block address

UDMA = Ultra DMA (Direct Memory Access)

ATA-1 (AT Attachment Interface for Disk Drives)

Although ATA-1 had been used since 1986 before being published as a standard, and although it was first published in 1988 in draft form, ATA-1 wasn't officially approved as a standard until 1994 (committees often work slowly). ATA-1 defined the original AT Attachment interface, which was an integrated bus interface between disk drives and host systems based on the ISA (AT) bus. The major features that were introduced and documented in the ATA-1 specification are

- 40/44-pin connectors and cabling
- Master/slave or cable select drive configuration options
- Signal timing for basic PIO (Programmed I/O) and DMA (Direct Memory Access) modes
- CHS (cylinder head sector) and LBA (logical block address) drive parameter translations supporting drive capacities up to $2^{28}–2^{20}$ (267,386,880) sectors, or 136.9GB

ATA-1 was officially published as ANSI X3.221-1994, AT Attachment Interface for Disk Drives, and was officially withdrawn on August 6, 1999. ATA-2 and later are considered backward-compatible replacements.

Although ATA-1 supported theoretical drive capacities up to 136.9GB ($2^{28}–2^{20}$ = 267,386,880 sectors), it did not address BIOS limitations that stopped at 528MB (1024×16×63 = 1,032,192 sectors). The BIOS limitations would be addressed in subsequent ATA versions because, at the time, no drives larger than 528MB had existed.

ATA-2 (AT Attachment Interface with Extensions-2)

Approved in 1996, ATA-2 was a major upgrade to the original ATA standard. Perhaps the biggest change was almost a philosophical one. ATA-2 was updated to define an interface between host systems and storage devices in general and not only disk drives. The major features added to ATA-2 as compared to the original ATA standard include

- Faster PIO and DMA transfer modes
- Support for power management
- Support for removable devices
- PCMCIA (PC card) device support
- Identify Drive command reports more information
- Defined standard CHS/LBA translation methods for drives up to 8.4GB in capacity

The most important additions in ATA-2 were the support for faster PIO and DMA modes, as well as methods to enable BIOS support up to 8.4GB. The BIOS support was necessary because, although even ATA-1 was designed to support drives of up to 136.9GB in capacity, the PC BIOS could originally handle drives of up to 528MB. Adding parameter-translation capability now allowed the BIOS to handle drives up to 8.4GB. This is discussed in more detail later in this chapter.

ATA-2 also featured improvements in the Identify Drive command that enabled a drive to tell the software exactly what its characteristics are; this is essential for both Plug and Play (PnP) and compatibility with future revisions of the standard.

ATA-2 was also known by unofficial marketing terms such as fast-ATA or fast-ATA-2 (Seagate/Quantum) and EIDE (Enhanced IDE, Western Digital). ATA-2 was officially published as ANSI X3.279-1996 AT Attachment Interface with Extensions.

ATA-3 (AT Attachment Interface-3)

First published in 1997, ATA-3 was a comparatively minor revision to the ATA-2 standard that preceded it. It consisted of a general cleanup of the specification and had mostly minor clarifications and revisions. The most major changes included the following:

- Eliminated single-word (8-bit) DMA transfer protocols.
- Added S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology) support for prediction of device performance degradation.
- LBA mode support was made mandatory (previously it had been optional).
- Added security mode, allowing password protection for device access.
- Recommendations for source and receiver bus termination to solve noise issues at higher transfer speeds.

ATA-3 has been officially published as ANSI X3.298-1997, AT Attachment 3 Interface.

ATA-3, which builds on ATA-2, adds improved reliability, especially of the faster PIO mode 4 transfers; however, ATA-3 does not define any faster modes. ATA-3 also adds a simple password-based security scheme, more sophisticated power management, and S.M.A.R.T. This enables a drive to keep track of problems that might result in a failure and therefore avoid data loss. S.M.A.R.T. is a reliability prediction technology that was initially developed by IBM.

ATA/ATAPI-4 (AT Attachment with Packet Interface Extension-4)

First published in 1998, ATA-4 included several important additions to the standard. It included the Packet Command feature known as the AT Attachment Packet Interface (ATAPI), which allowed devices such as CD-ROM and CD-RW drives, LS-120 SuperDisk floppy drives, tape drives, and other types of storage devices to be attached through a common interface. Until ATA-4 came out, ATAPI was a separately published standard. ATA-4 also added the 33MB/sec transfer mode known as Ultra-DMA or Ultra-ATA. ATA-4 is backward compatible with ATA-3 and earlier definitions of the ATAPI. The major revisions added in ATA-4 were as follows:

- Ultra-DMA (UDMA) transfer modes up to Mode 2, which is 33MB/sec (called UDMA/33 or Ultra-ATA/33)
- Integral ATAPI support
- Advanced power management support
- Defined an optional 80-conductor, 40-pin cable for improved noise resistance

- Compact Flash Adapter (CFA) support
- Introduced enhanced BIOS support for drives over 9.4ZB (zettabytes or trillion gigabytes) in size (even though ATA was still limited to 136.9GB)

ATA-4 was published as ANSI NCITS 317-1998, ATA-4 with Packet Interface Extension.

The speed and level of ATA support in your system is mainly dictated by your motherboard chipset. Most motherboard chipsets come with a component called either a South Bridge or an I/O Controller Hub that provides the ATA interface (as well as other functions) in the system. Check the specifications for your motherboard or chipset to see whether yours supports the faster ATA/33, ATA/66, or ATA/100 mode. One indication is to enter the BIOS Setup, put the hard disk on manual parameter settings (user defined), and see which if any Ultra-DMA modes are listed. Most boards built during 1998 support ATA/33; in 2000 they began to support ATA/66; and by late 2000 most started supporting ATA/100.

◀◀ See "Chipsets," p. 225.

ATA-4 made ATAPI support a full part of the ATA standard, and thus ATAPI was no longer an auxiliary interface to ATA but merged completely within. This promoted ATA for use as an interface for many other types of devices. ATA-4 also added support for new Ultra-DMA modes (also called Ultra-ATA) for even faster data transfer. The highest-performance mode, called UDMA/33, had 33MB/second bandwidth—twice that of the fastest programmed I/O mode or DMA mode previously supported. In addition to the higher transfer rate, because UDMA modes relieve the load on the processor, further performance gains were realized.

An optional 80-conductor cable (with cable select) is defined for UDMA/33 transfers. Although this cable was originally defined as optional, it would later be required for the faster ATA/66 and ATA/100 modes in ATA-5 and later.

Also included was support for queuing commands, which is similar to that provided in SCSI-2. This enabled better multitasking as multiple programs make requests for IDE transfers.

▶▶ See "SCSI Versus IDE," p. 540.

ATA/ATAPI-5 (AT Attachment with Packet Interface-5)

This version of the ATA standard was approved in early 2000 and builds on ATA-4. The major additions in the standard include the following:

- Ultra-DMA (UDMA) transfer modes up to Mode 4, which is 66MB/sec (called UDMA/66 or Ultra-ATA/66).
- 80-conductor cable now mandatory for UDMA/66 operation.
- Added automatic detection of 40- or 80-conductor cables.
- UDMA modes faster than UDMA/33 are enabled only if an 80-conductor cable is detected.

Another standard recently approved by the T13 committee is an IEEE-1394 (iLink or FireWire) extension to the ATA interface protocol. This specifies a bridge protocol between the iLink/FireWire bus and ATA, allowing ATA drives to be adapted to this newer interface.

▶▶ See "IEEE-1394," p. 948.

Copies of any of the published standards can be purchased from ANSI or Global Engineering Documents (see the Vendor List on the CD).

ATA-5 includes Ultra-ATA/66 (also called Ultra-DMA or UDMA/66), which doubles the Ultra-ATA burst transfer rate by reducing setup times and increasing the clock rate. The faster clock rate increases interference, which causes problems with the standard 40-pin cable used by ATA and Ultra-ATA. To eliminate noise and interference, the new 40-pin, 80-conductor cable has now been made mandatory for drives running in UDMA/66 or faster modes. This cable was first announced in ATA-4 but is now mandatory in ATA-5 to support the Ultra-ATA/66 mode. This cable adds 40 additional ground lines between each of the original 40 ground and signal lines, which help shield the signals from interference. Note that this cable works with older non–Ultra-ATA devices as well because it still has the same 40-pin connectors.

The 40-pin, 80-conductor cables will support the cable select feature and have color-coded connectors. The blue (end) connector should be connected to the ATA host interface (usually the motherboard). The black (opposite end) connector is known as the *master position*, which is where the primary drive plugs in. The gray (middle) connector is for slave devices.

To use either the UDMA/33 or UDMA/66 mode, your ATA interface, drive, BIOS, and cable must be capable of supporting the mode you want to use. The operating system also must be capable of handling Direct Memory Access. Windows 95 OSR2 or later, Windows 98/Me, and Windows 2000/XP are ready out of the box, but older versions of Windows 95 and NT (prior to Service Pack 3) require additional or updated drivers to fully exploit these faster modes.

For reliability, Ultra-DMA modes incorporate an error-detection mechanism known as *cyclical redundancy checking (CRC)*. CRC is an algorithm that calculates a checksum used to detect errors in a stream of data. Both the host (controller) and the drive calculate a CRC value for each Ultra-DMA transfer. After the data is sent, the drive calculates a CRC value, and this is compared to the original host CRC value. If a difference is reported, the host might be required to select a slower transfer mode and retry the original request for data.

ATA/ATAPI-6 (AT Attachment with Packet Interface-6)

This final version of the ATA standard was under development during 2001 and builds on ATA-5. The major additions in the standard include the following:

- Ultra-DMA (UDMA) transfer modes up to Mode 5, which is 100MB/sec (called UDMA/100, Ultra-ATA/100, or just ATA/100)
- LBA addressing extended to 2^{48} (281,474,976,710,656) sectors supporting drives up to 144.12PB (petabytes = quadrillion bytes)

ATA-6 includes Ultra-ATA/100 (also called Ultra-DMA or UDMA/100), which increases the Ultra-ATA burst transfer rate by reducing setup times and increasing the clock rate. As with ATA-5, the faster modes require the improved 80-conductor cable. Using the new ATA/100 mode requires both a drive and motherboard interface that supports that mode.

Besides adding the 100MB/sec UDMA Mode 5 transfer rate, ATA-6 also extends drive capacity greatly, and just in time. ATA-5 and earlier standards supported drives of up to only 136.9GB in capacity, which was becoming a limitation as larger drives are becoming available. At least one company has introduced 3 1/2-inch drives that are larger than that, but originally they could be released only in SCSI versions because SCSI didn't share the same limitation problems of ATA. With ATA-6, the drive limit has been expanded from (2^{28}–2^{20}) sectors to (2^{48}) sectors. What this means is that Logical Block Addressing used to use a 28-bit number, but with ATA-6 it now uses a 48-bit number instead. With 512 bytes per sector, this raises maximum supported drive capacity to 144.12PB. That is equal to more than 144.12 quadrillion bytes!

As a historical note, ATA-6 will likely be the last revision of the venerable parallel ATA standard. The future of ATA is called Serial ATA, and it is covered later in this chapter.

ATA Features

The ATA standards have gone a long way toward eliminating incompatibilities and problems with interfacing IDE drives to ISA/PCI bus systems. The ATA specifications define the signals on the 40-pin connector, the functions and timings of these signals, cable specifications, and so on. The following section lists some of the elements and functions defined by the ATA specification.

ATA I/O Connector

The ATA interface connector is normally a 40-pin header-type connector with pins spaced 0.1 inches (2.54mm) apart and generally is keyed to prevent the possibility of installing it upside down (see Figures 7.1 and 7.2). To create a keyed connector, the manufacturer generally removes pin 20 from the male connector and blocks pin 20 on the female cable connector, which prevents the user from installing the cable backward. Some cables also incorporate a protrusion on the top of the female cable connector, which fits into a notch in the shroud surrounding the mating male connector on the device. The use of keyed connectors and cables is highly recommended; plugging in an IDE cable backward normally won't cause any permanent damage. However, it can lock up the system and prevent it from running at all.

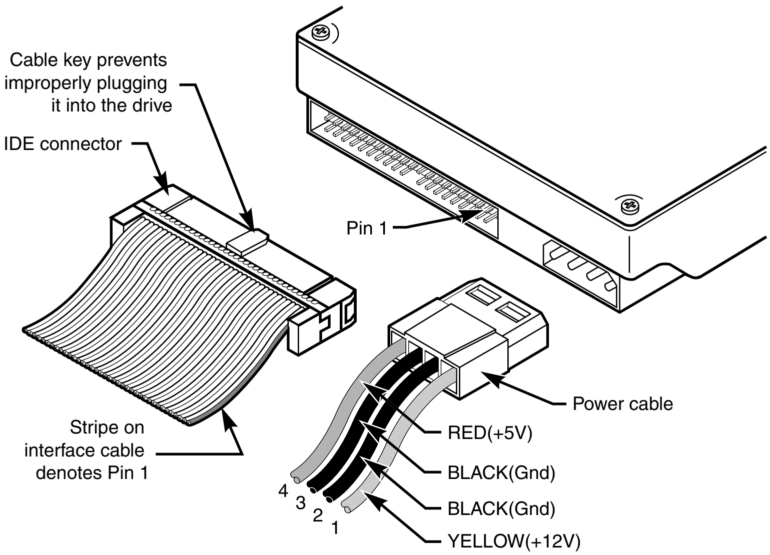

Figure 7.1 Typical ATA (IDE) hard drive connectors.

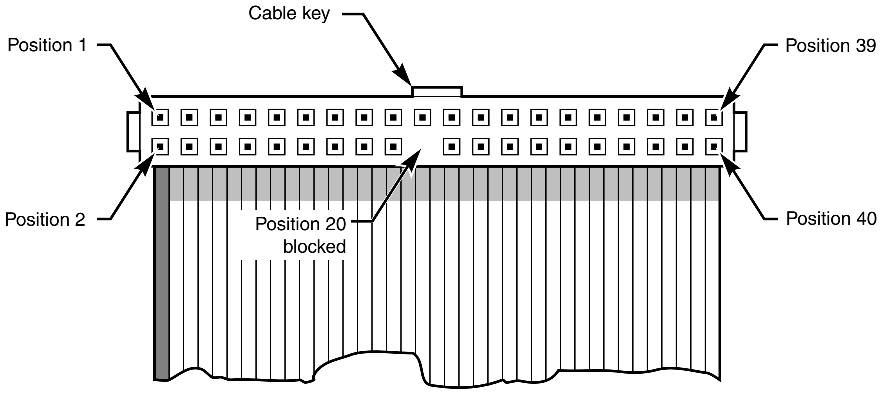

Figure 7.2 ATA (IDE) 40-pin interface connector detail.

Table 7.2 shows the standard 40-pin ATA-IDE interface connector pinout.

Table 7.2 40-Pin ATA Connector

Signal Name	Pin	Pin	Signal Name
-RESET	1	2	GROUND
Data Bit 7	3	4	Data Bit 8
Data Bit 6	5	6	Data Bit 9
Data Bit 5	7	8	Data Bit 10
Data Bit 4	9	10	Data Bit 11
Data Bit 3	11	12	Data Bit 12
Data Bit 2	13	14	Data Bit 13
Data Bit 1	15	16	Data Bit 14
Data Bit 0	17	18	Data Bit 15
GROUND	19	20	KEY (pin missing)
DRQ 3	21	22	GROUND
-IOW	23	24	GROUND
-IOR	25	26	GROUND
I/O CH RDY	27	28	CSEL:SPSYNC[1]
-DACK 3	29	30	GROUND
IRQ 14	31	32	Reserved[2]
Address Bit 1	33	34	-PDIAG
Address Bit 0	35	36	Address Bit 2
-CS1FX	37	38	-CS3FX
-DA/SP	39	40	GROUND
+5V (Logic)	41	42	+5V (Motor)
GROUND	43	44	Reserved

1. Pin 28 is normally cable select, but some older drives could use it for spindle synchronization.

2. Pin 32 was defined as -IOCS16 in ATA-2 but is no longer used.

Notebook-size, 2 1/2-inch drives normally use a smaller unitized 50-pin header connector with pins spaced only 2.0mm (0.079 inches) apart. The main 40-pin part of the connector is the same as the standard ATA connector (except for the physical pin spacing), but there are added pins for power and jumpering. Normally, the cable that plugs into this connector has 44 pins, carrying power as well as the standard ATA signals. The jumper pins normally have a jumper on them (the jumper position controls cable select, master, or slave settings). Figure 7.3 shows the unitized 50-pin connector used on 2 1/2-inch ATA drives.

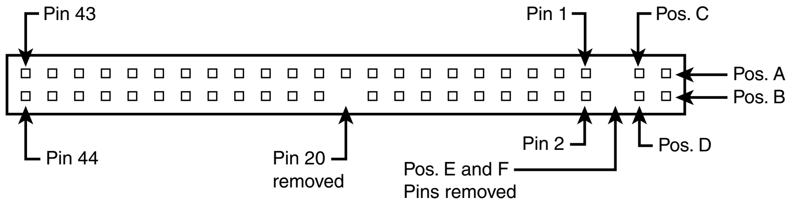

Figure 7.3 50-pin unitized ATA connector detail (used on 2 1/2-inch ATA drives).

Note the jumper pins at positions A–D and that the pins at positions E and F are removed. A jumper usually is placed between positions B and D to set the drive for cable select operation. On this connector pin 41 provides +5V power to the drive logic (circuit board), pin 42 provides +5V power to the motor (2 1/2-inch drives use 5V motors, unlike larger drives that normally use 12V motors), and pin 43 provides a power ground. The last pin (44) is reserved and not used.

Table 7.3 shows the 50-pin unitized ATA interface connector pinout as used on most 2 1/2-inch (laptop or notebook computer) drives.

Table 7.3 50-Pin Unitized ATA Connector Pinout

Signal Name	Pin	Pin	Signal Name
Jumper pin	A	B	Jumper pin
Jumper pin	C	D	Jumper pin
KEY (pin missing)	E	F	KEY (pin missing)
-RESET	1	2	GROUND
Data Bit 7	3	4	Data Bit 8
Data Bit 6	5	6	Data Bit 9
Data Bit 5	7	8	Data Bit 10
Data Bit 4	9	10	Data Bit 11
Data Bit 3	11	12	Data Bit 12
Data Bit 2	13	14	Data Bit 13
Data Bit 1	15	16	Data Bit 14
Data Bit 0	17	18	Data Bit 15
GROUND	19	20	KEY (pin missing)
DRQ 3	21	22	GROUND
-IOW	23	24	GROUND

Table 7.3 Continued

Signal Name	Pin	Pin	Signal Name
-IOR	25	26	GROUND
I/O CH RDY	27	28	CSEL
-DACK 3	29	30	GROUND
IRQ 14	31	32	Reserved
Address Bit 1	33	34	-PDIAG
Address Bit 0	35	36	Address Bit 2
-CS1FX	37	38	-CS3FX
-DA/SP	39	40	GROUND
+5V (Logic)	41	42	+5V (Motor)
GROUND	43	44	Reserved

Not All Cables and Connectors Are Keyed

Note that many lower-cost board and cable manufacturers leave out the keying. Cheaper motherboards often won't have pin 20 removed on their IDE connectors, and consequently they won't supply a cable with pin 20 blocked. If they don't use a shrouded connector with a notch and a corresponding protrusion on the cable connector, no keying exists and the cables can be inserted backward. Fortunately, the only consequence of this in most cases is that the device won't work until the cable is attached with the correct orientation.

In rare situations in which you are mixing and matching items, you might encounter a cable with pin 20 blocked (as it should be) and a board with pin 20 still present. In that case, you can break off pin 20 from the board, or for the more squeamish, remove the block from the cable. Some cables have the block permanently installed as a part of the connector housing, in which case you must break off pin 20 on the board or device end or use a different cable.

The simple rule of thumb is that pin 1 should be oriented toward the power connector on the device, which normally corresponds to the stripe on the cable.

ATA I/O Cable

A 40-conductor ribbon cable is specified to carry signals between the bus adapter circuits and the drive (controller). To maximize signal integrity and eliminate potential timing and noise problems, the cable should not be longer than 18 inches (0.46 meters).

Note that ATA drives supporting the higher-speed transfer modes, such as PIO Mode 4 or any of the Ultra-DMA (UDMA) modes, are especially susceptible to cable integrity problems and cables that are too long. If the cable is too long, you can experience data corruption and other errors that can be maddening. This will be manifested in any type of problem reading from or writing to the drive. In addition, any drive using UDMA Mode 4 (66MB/sec transfer rate) or Mode 5 (100MB/sec transfer rate) must use a special higher-quality 80-conductor cable (the extra conductors are grounds to reduce noise). I also recommend this type of cable if your drive is running at UDMA Mode 2 (33MB/sec) or slower because it can't hurt and can only help. I always keep a high-quality 80-conductor IDE cable in my toolbox for testing drives where I suspect cable integrity or cable length problems. Figure 7.4 shows the typical ATA cable layout and dimensions.

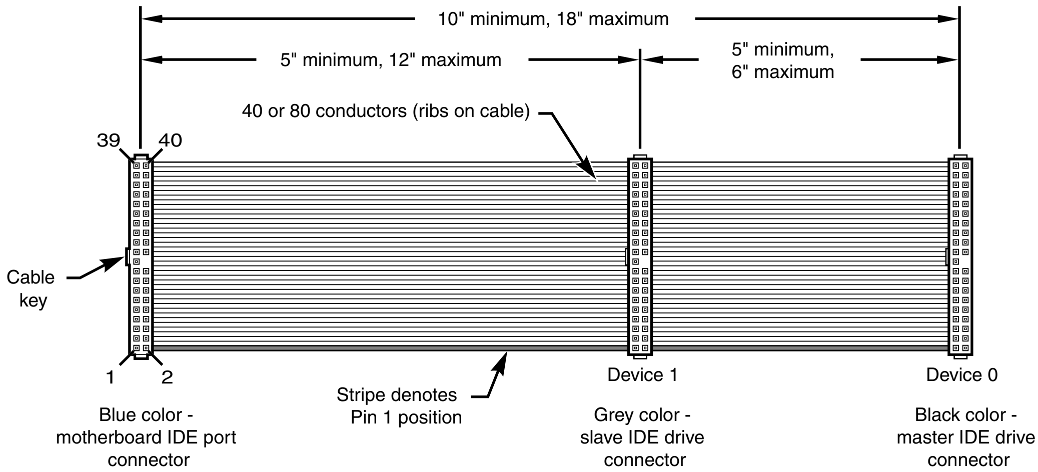

Figure 7.4 ATA (IDE) cable, with 40-pin connectors and either 40- or 80-conductor cables (additional wires are grounded in 80-conductor versions).

Note

Most 40-conductor cables do not color-code the connectors, whereas all 80-conductor cables do color-code the connectors.

Two primary variations of IDE cables are used today: one with 40 conductors and the other with 80 conductors. Both use 40-pin connectors, and the additional wires in the 80-conductor version are simply wired to ground. The additional conductors are designed to reduce noise and interference and are required when setting the interface to run at 66MB/sec (ATA/66) or faster. The drive and host adapter are designed to disable the higher-speed ATA/66 or ATA/100 modes if an 80-conductor cable is not detected. The 80-conductor cable can also be used at lower speeds; although this is unnecessary, it improves the signal integrity. Therefore, it is the recommended version no matter which drive you use.

Note the keying on the cable that is designed to prevent backward installation.

ATA Signals

This section describes some of the most important signals in more detail.

Pin 20 is used as a key pin for cable orientation and is not connected to the interface. This pin should be missing from any ATA connectors, and the cable should have the pin-20 hole in the connector plugged off to prevent the cable from being plugged in backward.

Pin 39 carries the drive active/slave present (DASP) signal, which is a dual-purpose, time-multiplexed signal. During power-on initialization, this signal indicates whether a slave drive is present on the interface. After that, each drive asserts the signal to indicate that it is active. Early drives could not multiplex these functions and required special jumper settings to work with other drives. Standardizing this function to allow for compatible dual-drive installations is one of the features of the ATA standard. This is why some drives require a slave present (SP) jumper, whereas others do not.

Pin 28 carries the cable select signal (CSEL). In some older drives, it could also carry a spindle synchronization signal (SPSYNC), but that is not commonly found on newer drives. The CSEL function is the most widely used and is designed to control the designation of a drive as master (drive 0) or slave (drive 1) without requiring jumper settings on the drives. If a drive sees the CSEL as being grounded, the drive is a master; if CSEL is open, the drive is a slave.

You can install special cabling to ground CSEL selectively. This installation usually is accomplished through a Y-cable arrangement, with the IDE bus connector in the middle and each drive at opposite ends of the cable. One leg of the Y has the CSEL line connected through, indicating a master drive; the other leg has the CSEL line open (conductor interrupted or removed), making the drive at that end the slave.

Dual-Drive Configurations

Dual-drive ATA installations can be problematic because each drive has its own controller and both controllers must function while being connected to the same bus. There has to be a way to ensure that only one of the two controllers will respond to a command at a time.

The ATA standard provides the option of operating on the AT bus with two drives in a daisy-chained configuration. The primary drive (drive 0) is called the master, and the secondary drive (drive 1) is called the slave. You designate a drive as being master or slave by setting a jumper or switch on the drive or by using a special line in the interface called the cable select pin and setting the CS jumper on the drive.

When only one drive is installed, the controller responds to all commands from the system. When two drives (and, therefore, two controllers) are installed, both controllers receive all commands from the system. Each controller then must be set up to respond only to commands for itself. In this situation, one controller must be designated as the master and the other as the slave. When the system sends a command for a specific drive, the controller on the other drive must remain silent while the selected controller and drive are functioning. Setting the jumper to master or slave enables discrimination between the two controllers by setting a special bit (the DRV bit) in the Drive/Head Register of a command block.

Configuring IDE drives can be simple, as is the case with most single-drive installations, or troublesome, especially when it comes to mixing two drives from different manufacturers on a single cable.

Most IDE drives can be configured with four possible settings:

- Master (single-drive)
- Master (dual-drive)
- Slave (dual-drive)
- Cable select

Some drives simplify this to three settings: master, slave, and cable select. Because each IDE drive has its own controller, you must specifically tell one drive to be the master and the other to be the slave. No functional difference exists between the two, except that the drive that's specified as the slave will assert a signal called DASP after a system reset informs the master that a slave drive is present in the system. The master drive then pays attention to the drive select line, which it otherwise ignores. Telling a drive that it's the slave also usually causes it to delay its spinup for several seconds to allow the master to get going and thus to lessen the load on the system's power supply.

Until the ATA IDE specification, no common implementation for drive configuration was in use. Some drive companies even used different master/slave methods for different models of drives. Because of these incompatibilities, some drives work together only in a specific master/slave or slave/master order. This situation mostly affects older IDE drives introduced before the ATA specification.

Most drives that fully follow the ATA specification now need only one jumper (master/slave) for configuration. A few also need a slave present jumper, as well. Table 7.4 shows the jumper settings required by most ATA IDE drives.

Table 7.4 Jumper Settings for Most ATA IDE-Compatible Drives on Standard (Non–Cable Select) Cables

Jumper Name	Single-Drive	Dual-Drive Master	Dual-Drive Slave
Master (M/S)	On	On	Off
Slave Present (SP)	Off	On	Off
Cable Select (CS)	Off	Off	Off

Note

If a cable select cable is used, the CS jumper should be set On and all others should be Off. The cable connector then determines which drive will be master or slave.

Figure 7.5 shows the jumpers on a typical ATA drive.

The master jumper indicates that the drive is a master or a slave. Some drives also require a slave present jumper, which is used only in a dual-drive setup and then installed only on the master drive—which is somewhat confusing. This jumper tells the master that a slave drive is attached. With many ATA IDE drives, the master jumper is optional and can be left off. Installing this jumper doesn't hurt in these cases and can eliminate confusion; I recommend that you install the jumpers listed here.

Note

Note that some drives have these jumpers on the drive circuit board, and as such they might not be visible on the rear.

To eliminate confusion over master/slave settings, most newer systems now use the cable select option. This involves two things. The first is having a special IDE cable that has all the wires except pin 28 running from the motherboard connector to both drive connectors. Pin 28 is used for cable select and is connected to one of the drive connectors (labeled master) and not to the other (labeled slave). Both drives are then configured in cable select mode via the CS jumper on each drive.

With cable select, the drive that receives signals on pin 28 automatically becomes the master, and the other becomes the slave. Most cables implement this by removing the metal insulation displacement bit from the pin-28 hole, which can be difficult to see at a glance. Other cables have a section of pin 28 visibly cut from the cable somewhere along the ribbon. Because this is such a minor modification to the cable and can be difficult to see, cable select cables normally have the connectors labeled master, slave, and system, indicating that the cable controls these options rather than the drive. All 80-conductor UltraATA cables are designed to use cable select.

With cable select, you simply set the CS jumper on all drives and then plug the drive you want to be the master into the connector labeled master on the cable and the drive you want to be the slave into the connector labeled slave.

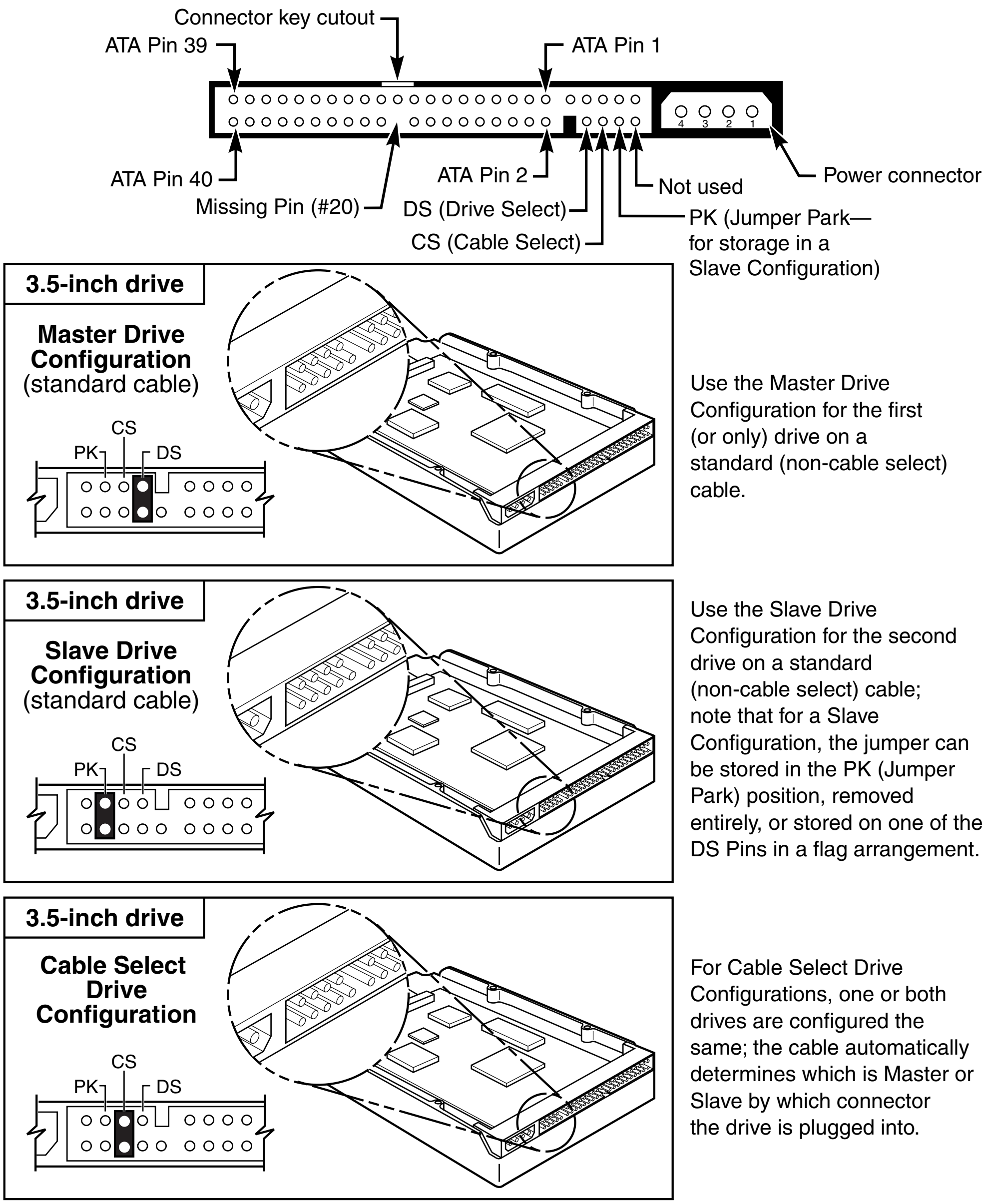

Figure 7.5 ATA (IDE) drive jumpers for most drives. Note that the drive is shown upside down in this figure, whereas previous figures in this chapter depict drives as they appear right side up.

ATA Commands

One of the best features of the ATA IDE interface is the enhanced command set. The ATA IDE interface was modeled after the WD1003 controller IBM used in the original AT system. All ATA IDE drives must support the original WD command set (eight commands) with no exceptions, which is why IDE drives are so easy to install in systems today. All IBM-compatible systems have built-in ROM BIOS support for the WD1003, so they essentially support ATA IDE as well.

In addition to supporting all the WD1003 commands, the ATA specification added numerous other commands to enhance performance and capabilities. These commands are an optional part of the ATA interface, but several of them are used in most drives available today and are very important to the performance and use of ATA drives in general.

Perhaps the most important is the Identify Drive command. This command causes the drive to transmit a 512-byte block of data that provides all details about the drive. Through this command, any program (including the system BIOS) can find out exactly which type of drive is connected, including the drive manufacturer, model number, operating parameters, and even the serial number of the drive. Many modern BIOSes use this information to automatically receive and enter the drive's parameters into CMOS memory, eliminating the need for the user to enter these parameters manually during system configuration. This arrangement helps prevent mistakes that can later lead to data loss when the user no longer remembers what parameters he used during setup.

The Identify Drive data can tell you many things about your drive, including the following:

- Number of cylinders in the recommended (default) translation mode
- Number of heads in the recommended (default) translation mode
- Number of sectors per track in the recommended (default) translation mode
- Number of cylinders in the current translation mode
- Number of heads in the current translation mode
- Number of sectors per track in the current translation mode
- Manufacturer and model number
- Firmware revision
- Serial number
- Buffer type, indicating sector buffering or caching capabilities

Several public-domain programs can execute this command to the drive and report the information onscreen. I use the IDEINFO (available at `http://www.dc.ee/Files/Utils/IDEINFO.ARJ`) or IDEDIAG (available from many of the popular shareware sites) program. I find these programs especially useful when I am trying to install IDE drives and need to know the correct parameters for a user-definable BIOS type. These programs get the information directly from the drive.

Two other important commands are the Read Multiple and Write Multiple commands. These commands permit multiple-sector data transfers and, when combined with block-mode PIO capabilities in the system, can result in incredible data-transfer rates many times faster than single-sector PIO transfers.

MMMany other enhanced commands are available, including room for a given drive manufacturer to implement what are called vendor-unique commands. These commands often are used by a particular vendor for features unique to that vendor. Often, features such as low-level formatting and defect management are controlled by vendor-unique commands. This is why low-level format programs can be so specific to a particular manufacturer's IDE drives and why many manufacturers make their own LLF programs available.

ATA Upgrades

Since ATA-1, newer versions of the ATA interface and complementary BIOS support larger and faster drives, as well as different types of devices other than hard disks. This section details these areas of improvement.

ATA-2 through ATA-6 have improved the original ATA/IDE interface in four main areas:

- Secondary two-device channel
- Increased maximum drive capacity
- Faster data transfer
- ATAPI (ATA Program Interface)

The following sections describe these improvements.

Secondary ATA Channel

Although even ATA-1 was never limited to a single channel, systems at that time normally had only a primary channel built into the motherboard. Starting when the ATA-2 standard was published, most systems began including a second ATA interface, called the secondary channel. Because each channel could support two drives, this allowed for up to four devices to be installed.

If more than four ATA devices are needed, you can install additional ATA ports via add-on cards, which normally plug into PCI slots. These additional ports are not supported directly by the motherboard BIOS, but the card can have an adapter BIOS that will be recognized at boot time. Even without an onboard BIOS, devices attached to the card can be supported by drivers—the only feature lacking without ROM-based support is that the capability to be directly bootable.

Companies such as Promise Technologies, MicroFirmware, GSI, and others make add-on ATA adapters that allow using more than the two ports that come built into most motherboards.

Drive Capacity Limitations

ATA interface versions up through ATA-5 suffered from a drive capacity limitation of 136.9GB (billion bytes). Depending on the BIOS used, this limitation can be further reduced to 8.4GB, or even as low as 528MB (million bytes). This is due to limitations in both the BIOS and the ATA interface, which when combined create even further limitations. To understand these limits, you have to look at the BIOS and ATA interface.

The pertinent limitations are those of the ATA interface itself as well as the BIOS (drivers) used to talk to the interface. A summary of the limitations are shown in Table 7.5.

Table 7.5 ATA/IDE Drive Capacity Limitations

Specification	Maximum Sectors	Capacity[1]	Maximum Capacity
ATA-5 interface	267,386,880	136,902,082,560	136.9GB
Standard CHS BIOS	1,032,192	528,482,304	528.5MB
CHS BIOS with translation	16,515,072	8,455,716,864	8.4GB
Enhanced (EDD BIOS)	18,446,744,073,709,551,600	9,444,732,965,739,290,430,000	9.4ZB

1. Maximum capacity in bytes
ATA = AT Attachment
CHS = cylinder head sector
EDD = enhanced disk drive
MB = million bytes
GB = billion bytes
ZB = zettabytes, one ZB equals one billion trillion bytes

This section explores these limitations in more detail.

The BIOS-based driver for hard disks is accessed via software interrupt 13h (13 hex), which offers functions for reading and writing drives at the sector level. INT13h requires that the particular sector be addressed by its cylinder, head, and sector location, otherwise known as *CHS addressing*. This interface is used by the operating system and low-level disk utilities to access the drive. The INT13h interface originally was written by IBM for the BIOS on the PC XT hard disk controller in 1983 and then incorporated into the AT motherboard BIOS in 1984. This interface used numbers to define the particular cylinder, head, and sector being addressed. Table 7.6, which shows the standard INT13h BIOS CHS parameter limits, includes the maximum values for these numbers.

Table 7.6 INT13h BIOS CHS Parameter Limits

Field	Field Size	Maximum Value	Range	Total Usable
Cylinder	10 bits	1,024	0–1023	1,024
Head	8 bits	256	0–255	256
Sector	6 bits	64	1–63	63

The concept is simple: If you had a hotel with two-digit decimal numbers, you could have only 100 rooms, numbered 0–99. CHS numbers used by the BIOS INT13h interface are binary, and with a 10-bit number being used for cylinders, you can have only 1,024 maximum, numbered 0–1023. Because the head you want to access is identified by an 8-bit number, the maximum number of heads the BIOS could handle is 256, numbered 0–255. Finally, with sectors there is a minor difference. Sectors on a track are identified by a 6-bit number, which would normally allow a maximum of 64; because sectors are numbered starting with 1 (instead of 0), however, the range is limited to 1–63, which means a total of 63 sectors per track is the maximum the BIOS can handle.

These BIOS limitations are true for all BIOS versions or programs that rely on standard CHS addressing, using the INT13h BIOS interface. Using the maximum numbers possible for CHS, a drive with 1,024 cylinders, 256 heads, and 63 sectors per track is possible. Because each sector is 512 bytes, the math works out as follows:

```
                Max. Values
---------------------------
    Cylinders         1,024
        Heads           256
```

```
Sectors/Track           63
==========================
Total Sectors   16,515,072
--------------------------
Total Bytes  8,455,716,864
Megabytes (MB)       8,456
Mebibytes (MiB)      8,064
Gigabytes (GB)         8.4
Gibibytes (GiB)        7.8
```

From these calculations, you can see that the maximum size drive addressable via the BIOS INT13h interface is about 8.4GB (where GB equals roughly one billion bytes), or 7.8GiB (where GiB means *gigabinarybytes*).

Note that the Gi (gigabinary) designation is a standard designed to eliminate confusion between decimal- and binary-based multiples, especially in computer systems. In December 1998, the International Electrotechnical Commission (IEC) approved as an international standard the prefix names and symbols for binary multiples used in data processing and transmission. The prefixes are shown in Table 7.7.

Table 7.7 Standard Prefix Names and Symbols for Binary Multiples

			Decimal Prefixes	
Factor	**Symbol**	**Name**	**Value**	
10^3	K	Kilo	1,000	
10^6	M	Mega	1,000,000	
10^9	G	Giga	1,000,000,000	
10^{12}	T	Tera	1,000,000,000,000	
10^{15}	P	Peta	1,000,000,000,000,000	
			Binary Prefixes	
Factor	**Symbol**	**Name**	**Derivation**	**Value**
2^{10}	Ki	Kibi	Kilobinary	1,024
2^{20}	Mi	Mebi	Megabinary	1,048,576
2^{30}	Gi	Gibi	Gigabinary	1,073,741,824
2^{40}	Ti	Tebi	Terabinary	1,099,511,627,776
2^{50}	Pi	Pebi	Petabinary	1,125,899,906,842,624

Note

Under this standard terminology, an MB (megabyte) would be 1,000,000 bytes, whereas an MiB (mebibyte) would be 1,048,576 bytes.

Another way of looking at this is that each sector can be addressed using a single 24-bit number (10 bits plus 8 bits plus 6 bits). Because the 6-bit portion cannot be all 0s, the total number of sectors addressable would be defined as $2^{24} - 2^{18}$, or 16,777,216 – 262,144 = 16,515,072 sectors. Because each sector is 512 bytes, the maximum capacity supported by the BIOS is 8.4GB.

Unfortunately for ATA, the BIOS limits are not the only limitations that apply. Limits also exist in the ATA interface itself. The ATA limits are shown in Table 7.8.

Table 7.8 Standard ATA Parameter Limitations

Field	Field Size	Maximum Value	Range	Total Usable
Cylinder	16 bits	65536	0–65535	65536
Head	4 bits	16	0–15	16
Sector	8 bits	256	1–255	255

As you can see, the ATA limits are higher than the BIOS limits for cylinders and sectors but lower than the BIOS limits for heads. The total limits for capacity according to the ATA specifications are as follows:

```
                  Max. Values
-----------------------------
    Cylinders          65,536
        Heads              16
Sectors/Track             255
=============================
Total Sectors     267,386,880
-----------------------------
Total Bytes   136,902,082,560
Megabytes (MB)        136,902
Mebibytes (MiB)       130,560
Gigabytes (GB)          136.9
Gibibytes (GiB)         127.5
```

Note that the ATA limitations also can be expressed as a single 28-bit number—the 16-bit cylinder number plus the 4-bit head number plus the 8-bit sector number. To maintain compatibility between LBA and CHS modes in which the sector number must start from 1 instead of 0, the maximum capacity is $2^{28} - 2^{20}$, or 268,435,456 – 1,048,576 = 267,386,880 sectors. Because each sector is 512 bytes, the maximum capacity supported by ATA through ATA-5 is 136.9GB.

ATA-6 increases this limit substantially by going to an LBA interface that uses 48-bit sector addressing. This means that maximum capacity is increased to 2^{48} sectors. This is equal to 281,474,976,710,656 total sectors. Because each sector stores 512 bytes, this results in a maximum drive capacity of

```
                           Max. Values
--------------------------------------
Total Sectors      281,474,976,710,656
--------------------------------------
Total Bytes    144,115,188,075,855,888
Megabytes (MB)         144,115,188,076
Mebibytes (MiB)        137,438,953,472
Gigabytes (GB)             144,115,188
Gibibytes (GiB)            134,217,728
Terabytes (TB)                 144,115
Tebibytes (TiB)                131,072
Petabytes (PB)                     144
Pebibytes (PiB)                    128
```

This allows a capacity of just over 144PB (petabytes = quadrillion bytes)! It should be awhile before you have to worry about that barrier being broken.

When you combine the limitations of the BIOS and ATA interface, you end up with the situation as shown in Table 7.9.

Table 7.9 Combined CHS BIOS and ATA Parameter Limits

Field	CHS BIOS Parameter Limits	ATA Parameter Limits	Combined Limits
Cylinder	1,024	65,536	1,024
Head	256	16	16
Sector	63	255	63
Total Sectors	16,515,072	267,386,880	1,032,192
Maximum Capacity	8.4GB	136.9GB	528MB

As you can see, combining these limits results in maximum usable parameters of 1,024 cylinders, 16 heads, and 63 sectors, which results in a maximum drive capacity of 528MB.

Starting in 1993–94, most BIOSes began implementing a fix for these problems, which enabled drives up to the BIOS limit of 8.4GB to be used. This fix involved what is termed *parameter translation* at the BIOS Setup level, which adapted or translated the cylinder, head, and sector numbers to fit within the allowable BIOS parameters. There are two types of translation: One works mathematically off the reported CHS parameters (called Large or Extended CHS in the BIOS Setup); the other is derived from the total number of sectors (called Logical Block Address or LBA in the BIOS Setup). Both translation schemes result in the same thing; they just use two different mathematical algorithms to get there. Due to some limitations with the CHS translations, you usually should select LBA translation if that is an option in your BIOS setup.

Virtually all PC BIOSes since 1994 have translation capability in the BIOS Setup, and virtually all offer both CHS translation as well as LBA translation modes. If both translation modes are offered, you should choose the LBA method of translation because it is the more efficient of the two. Both normally result in the same translated CHS parameters; however, depending on the specific drive parameters reported, the geometries can vary depending on the translation method selected. If you were to set up and format a drive using CHS translation and then change to LBA translation, the interpreted geometry could change and the drive could then become unreadable. Bottom line: After you select a translation method, don't plan on changing it unless you have your data securely backed up.

With translation, the parameters reported by the drive are translated. The physical parameters reported by the drive are modified into parameters that are acceptable to the BIOS. Here is an example:

```
                     Physical      Logical
                          CHS          CHS
                   Parameters   Parameters
--------------------------------------------
        Cylinders      12,000          750
            Heads          16          256
    Sectors/Track          63           63
============================================
    Total Sectors  12,096,000   12,096,000
     Capacity (MB)      6,193        6,193
   Capacity (Meg)       5,906        5,906
```

This example shows a drive with 12,000 cylinders and 16 heads. The physical cylinder count is way above the BIOS limit of 1,024, so in translating the BIOS Setup divides the cylinder count by 2, 4, 8, or 16 to bring it down below 1,024. In this case it was necessary to divide by 16, which results in a new cylinder count of 750—well below the 1,024 maximum. Because the cylinder count was divided by 16, the head count is then multiplied by the same number, resulting in 256 heads, which is right at the limit the BIOS can handle. So, although the drive physically has 12,000 cylinders and 16 heads, the BIOS and all software (including the operating system) instead see the drive as having 750 cylinders and 256 heads. Note that the 63 sector count is simply carried over without change. The result is that by using the new parameters the BIOS can see the entire 6GB drive and won't be limited to just the first 528MB.

When you install a drive, you don't have to do the translation math; the BIOS Setup does that for you behind the scenes. All you have to do is either allow the BIOS to autodetect the drive physical parameters and enable ECHS (large) or LBA translation. The BIOS Setup does the rest of the work for you.

You usually can tell whether your BIOS supports translation by the capability to specify more than 1,024 cylinders in the BIOS Setup, although this can be misleading. The best clue is to look for the translation setting parameters in the IDE drive setup page in the BIOS Setup. See Chapter 5, "BIOS," for more information on how to enter the BIOS Setup on your system. If you see drive-related settings, such as LBA or ECHS (sometimes called Large or Extended), these are telltale signs of a BIOS with translation support. Most BIOSes with a date of 1994 or later include this capability. If your system currently does not support parameter translation, you might be able to get an upgrade from your motherboard manufacturer or install a BIOS upgrade card with this capability such as the ATA Pro Flash by MicroFirmware (see the Vendor List on the CD included with this book).

Table 7.10 shows the four ways today's BIOSes can handle addressing sectors on the drive: Standard CHS (no translation), Extended CHS translation, LBA translation, and pure LBA addressing. They are summarized in the following table.

Table 7.10 Drive Sector Addressing Methods

BIOS Mode	Operating System to BIOS	BIOS to Drive
Standard (normal) No translation	Physical CHS Parameters	Physical CHS Parameters
Extended CHS (large) Translation	Logical CHS Parameters	Physical CHS Parameters
LBA Translation	Logical CHS Parameters	LBA Parameters
Pure LBA (EDD BIOS)	LBA Parameters	LBA Parameters

In Standard CHS, there is only one possible translation step internal to the drive. The drive's actual physical geometry is completely invisible from the outside with all zoned recorded ATA drives today. The cylinders, heads, and sectors printed on the label for use in the BIOS setup are purely logical geometry and do not represent the actual physical parameters. Standard CHS addressing is limited to 16 heads and 1,024 cylinders, which provides a limit of 504MB.

This is often called "Normal" in the BIOS setup and causes the BIOS to behave like an old-fashioned one without translation. Use this setting if your drive has fewer than 1,024 cylinders or if you want to use the drive with a non-DOS operating system that doesn't understand translation.

In Extended CHS, a translated logical geometry is used to communicate between the drive and the BIOS, whereas a different translated geometry is used to communicate between the BIOS and everything else. In other words, normally two translation steps exist. The drive still translates internally but has logical parameters that exceed the 1,024-cylinder limitation of the standard BIOS. In this case, the drive's cylinder count is usually divided by 2, and the head count is multiplied by 2 to get the translated values from those actually stored in the CMOS Setup. This type of setting breaks the 504MB/528MB barrier.

This often is called "Large" or "ECHS" in the BIOS setup and tells the BIOS to use Extended CHS translation. It uses a different geometry (cylinders/heads/sectors) when talking to the drive than when talking to the BIOS. This type of translation should be used with drives that have more than 1,024 cylinders but that do not support LBA. Note that the geometry entered in your BIOS Setup is the logical geometry, not the translated one.

LBA is a means of linearly addressing sector addresses, beginning at Cylinder 0, Head 0, Sector 1 as LBA 0, and proceeding on to the last physical sector on the drive. This is new in ATA-2 but has always been the one-and-only addressing mode in SCSI.

With LBA, each sector on the drive is numbered starting from 0. The number is a 28-bit binary number internally, which translates to a sector number from 0 to 267,386,879. Because each sector represents 512 bytes, this results in a maximum drive capacity of exactly 136.9GB. Unfortunately, the operating system still needs to see a translated CHS, so the BIOS determines how many sectors there are and comes up with Translated CHS to match. The BIOS CHS limits are 1,024 cylinders, 256 heads, and 63 sectors per track, which limits total drive capacity to 8.4GB.

In other words, this scheme breaks the 528MB barrier in essentially the same way as Extended CHS does. Because it is somewhat simpler to use a single linear number to address a sector on the hard disk compared to a CHS type address, LBA translation is the preferred method if the drive supports it.

Caution

A word of warning with these BIOS translation settings: If you switch between Standard CHS, Extended CHS, or LBA, the BIOS can change the (translated) geometry. This usually occurs only if the number of sectors reported by the Identify Drive command is fewer than 63 because LBA translations result in 63 sectors per track. The same thing can happen if you transfer a disk that has been formatted on an old, non-LBA computer to a new one that uses LBA. This causes the logical CHS geometry seen by the operating system to change and the data to appear in the wrong locations from where it actually is! This can cause you to lose access to your data if you are not careful. I always recommend recording the CMOS Setup screens associated with the hard disk configuration so you can properly match the setup of a drive to the settings to which it was originally set.

Breaking the 8.4GB Barrier

Systems using ATA drives have been plagued by size limitations because of ATA interface and BIOS issues. The first two limits were mostly BIOS related; the first was at 528MB and the second at 8.4GB using parameter translation.

Unfortunately, translating parameters works only within the limitations of the BIOS, which means that, although translation breaks the 528MB barrier, it runs into another barrier at 8.4GB. Therefore, translation is still limited by the BIOS maximum of 8.4GB support.

Supporting drives larger than that requires a whole new addressing scheme, which no longer uses the clumsy CHS numbers and instead uses only the LBA or sector number directly. Phoenix Technologies recognized this problem early on and, beginning in 1994, began publishing a document titled "BIOS Enhanced Disk Drive Specification," which addressed this problem with an elegant solution. Their idea in publishing was to get the other BIOS manufacturers to follow their lead so all the BIOSes would be compatible with each other.

To ensure further development and compatibility, after several revisions in 1996, Phoenix turned this document over to the National Committee on Information Technology Standards (NCITS) for further enhancement and certification as a standard called the "BIOS Enhanced Disk Drive Specification (EDD)." Starting in 1998, most of the other BIOS manufacturers began installing EDD support in their BIOSes, finally enabling support for ATA drives larger than 8.4GB. Coincidentally (or not), this support arrived just in time because ATA drives of that size and larger became available that year.

The EDD document describes new services provided by the BIOS to support storage devices of up to 2^{64} sectors, which results in a capacity of more than 9.44ZB (a zettabyte equals a trillion GB: 9.44×10^{21} or, to be more precise, 9,444,732,965,739,290,430,000 bytes!). Older BIOS services have a compatibility limit of 528MB and a theoretical limit of 8.4GB. Starting in mid-1998, most systems incorporated the enhanced BIOS services. Note that even though the BIOS can handle drives with up to 2^{64} sectors, an ATA drive is still currently limited in size to 2^{48} sectors, or 144,115,188GB (144PB) maximum.

According to this EDD specification, drives are accessed by the pure LBA number without any CHS translation, while still preserving limited backward compatibility with CHS for the first 8.4GB of the drive. EDD has support for up to 2^{64} total sectors. This works out to a drive capacity as shown in the following calculations:

$2^{64} = 1.8446744073709551 6\times10^{19}$ sectors

$= 9.44473296573929043\times10^{21}$ bytes

= 9.4 giga-tera (billion trillion) bytes!

Phoenix originally claimed that the EDD specification would hold us for another 15 years. However, at the rate of growth where drive capacity doubles every 1.5–2 years (Moore's Law), considering that 180GB ATA drives were available in 2001, this standard should take us at least through the year 2055, and possibly as far as the year 2073 before we need a new address scheme with more bits.

If your system BIOS dates to 1998 or later, you most likely have EDD support, which means support for drives as large as 136.9GB.

With the ATA-6 specification in 2001, a new 48-bit address scheme has been designed that extends the limits to 2^{48} sectors, which is 144.12PB (petabytes). Because the BIOS services allow up to 2^{64} sectors, the 144 petabyte limitation will be the lower of the two that will apply. Still, that should hold us for some time to come.

Note that if you use older software including utilities, applications, or even operating systems that rely exclusively on CHS parameters, they will see all drives over 8.4GB as 8.4GB only. You will need not only a newer BIOS, but also newer software designed to handle the direct LBA addressing to work with drives over 8.4GB.

Operating systems limitations with respect to drives over 8.4GB are shown in Table 7.11.

Table 7.11 Operating Systems Limitations

Operating System	Limitations for Hard Drive Size
DOS/Windows 3x	DOS 6.22 or lower cannot support drives greater than 8.4GB. DOS 7.0 or higher (included with Windows 95 or later) is required to recognize a drive over 8.4GB.
Windows 9X/Me	Windows 95a (original version) does support the INT13h extensions, which means it does support drives over 8.4GB; however, due to limitations of the FAT16 file system, the maximum individual partition size is limited to 2GB. Windows 95B / OSR2 or later (including Windows 98/Me) supports the INT13h extensions, which allows drives over 8.4GB, and also supports FAT32, which allows partition sizes up to the maximum capacity of the drive.

Table 7.11 Continued

Operating System	Limitations for Hard Drive Size
Windows NT	Windows NT 3.5x does not support drives greater than 8.4GB. Windows NT 4.0 does support drivers greater than 8.4GB; however, when a drive larger than 8.4GB is being used as the primary bootable device, Windows NT will not recognize more than 8.4GB. Microsoft has released Service Pack 4, which corrects this problem.
Windows 2000/XP	Windows 2000/XP supports drives greater than 8.4GB.
OS/2 Warp	Some versions of OS/2 are limited to a boot partition size of 3.1GB or 4.3GB. IBM has a Device Driver Pack upgrade that enables the boot partition to be as large as 8.4GB. The HPFS file system in OS/2 will support drives up to 64GB.
Novell	NetWare 5.0 or later supports drives greater than 8.4GB.

Faster Data Transfer

ATA-2/EIDE and ATA-3 define several high-performance modes for transferring data to and from the drive. These faster modes are the main part of the new specifications and were the main reason they were initially developed. The following section discusses these modes.

The PIO mode determines how fast data is transferred to and from the drive. In the slowest possible mode—PIO mode 0—the data cycle time cannot exceed 600 nanoseconds (ns). In a single cycle, 16 bits are transferred into or out of the drive, making the theoretical transfer rate of PIO Mode 0 (600ns cycle time) 3.3MB/sec. Most of the high-performance ATA-2 (EIDE) drives today support PIO Mode 4, which offers a 16.6MB/sec transfer rate.

Table 7.12 shows the PIO modes, with their respective transfer rates.

Table 7.12 PIO (Programmed I/O) Modes and Transfer Rates

PIO Mode	Bus Width (bits)	Cycle Speed (ns)	Bus Speed (MHz)	Cycles per Clock	Transfer Rate (MB/sec)	ATA Specification
0	16	600	1.67	1	3.33	ATA-1
1	16	383	2.61	1	5.22	ATA-1
2	16	240	4.17	1	8.33	ATA-1
3	16	180	5.56	1	11.11	ATA-2
4	16	120	8.33	1	16.67	ATA-2

ATA-2 was also referred to as EIDE (Enhanced IDE) or Fast-ATA

ns = nanoseconds (billionths of a second)

MB = million bytes

To run in Mode 3 or 4 requires that the IDE port on the system be a local bus port. This means that it must operate through either a VL-Bus or PCI bus connection. Most motherboards with ATA-2/EIDE support have dual IDE connectors on the motherboard, and most of them now allow full throughput. Most of the motherboard chipsets include the ATA interface in their South Bridge components, which in modern systems is tied into the PCI bus.

Older 486 and some early Pentium boards have only the primary connector running through the system's PCI local bus. The secondary connector on those boards usually runs through the ISA bus and therefore supports up to Mode 2 operation only.

When interrogated with an Identify Drive command, a hard disk returns, among other things, information about the PIO and DMA modes it is capable of using. Most enhanced BIOSes automatically set the correct mode to match the capabilities of the drive. If you set a mode faster than the drive can handle, data corruption results.

ATA-2 and newer drives also perform Block Mode PIO, which means they use the Read/Write Multiple commands that greatly reduce the number of interrupts sent to the host processor. This lowers the overhead, and the resulting transfers are even faster.

DMA Transfer Modes

ATA drives also support *Direct Memory Access (DMA)* transfers. DMA means that the data is transferred directly between drive and memory without using the CPU as an intermediary, as opposed to PIO. This has the effect of offloading much of the work of transferring data from the processor, in effect allowing the processor to do other things while the transfer is taking place.

There are two distinct types of direct memory access: singleword (8-bit) and multiword (16-bit) DMA. Singleword DMA modes were removed from the ATA-3 and later specifications and are obsolete. DMA modes are also sometimes called *busmaster* ATA modes because they use a host adapter that supports busmastering. Ordinary DMA relies on the legacy DMA controller on the motherboard to perform the complex task of arbitration, grabbing the system bus and transferring the data. In the case of busmastering DMA, all this is done by a higher-speed logic chip in the host adapter interface (which is also on the motherboard).

Systems using the Intel PIIX (PCI IDE ISA eXcelerator) and later South Bridge chips have the capability of supporting busmaster IDE. The singleword and doubleword busmaster IDE modes and transfer rates are shown in Tables 7.13 and 7.14.

Table 7.13 Singleword (8-bit) DMA Modes and Transfer Rates

8-bit DMA Mode	Bus Width (bits)	Cycle Speed (ns)	Bus Speed (MHz)	Cycles per Clock	Transfer Rate (MB/sec)	ATA Specification
0	16	960	1.04	1	2.08	ATA-1[1]
1	16	480	2.08	1	4.17	ATA-1[1]
2	16	240	4.17	1	8.33	ATA-1[1]

1. Singleword (8-bit) DMA modes were removed from the ATA-3 and later specifications.

Table 7.14 Multiword (16-bit) DMA Modes and Transfer Rates

16-bit DMA Mode	Bus Width (bits)	Cycle Speed (ns)	Bus Speed (MHz)	Cycles per Clock	Transfer Rate (MB/sec)	ATA Specification
0	16	480	2.08	1	4.17	ATA-1
1	16	150	6.67	1	13.33	ATA-2
2	16	120	8.33	1	16.67	ATA-2

ATA-2 was also referred to as EIDE (Enhanced IDE) or Fast-ATA.

Note that doubleword DMA modes are also called busmaster DMA modes by some manufacturers. Unfortunately, even the fastest doubleword DMA Mode 2 results in the same 16.67MB/sec transfer speed as PIO Mode 4, so DMA modes have never really caught on as being desirable. However, even though the transfer speed is the same as PIO because DMA offloads much of the work from the processor, overall system performance would be higher. Even so, multiword DMA modes were never very popular and have been superseded by the newer Ultra-DMA modes supported in ATA-4–compatible devices.

Table 7.15 shows the Ultra-DMA modes now supported in the ATA-4 and ATA-5 specifications.

Table 7.15 Ultra-DMA Support in ATA-4 and ATA-5

Ultra DMA Mode	Bus Width (bits)	Cycle Speed (ns)	Bus Speed (MHz)	Cycles per Clock	Transfer Rate (MB/sec)	ATA Specification
0	16	240	4.17	2	16.67	ATA-4
1	16	160	6.25	2	25.00	ATA-4
2	16	120	8.33	2	33.33	ATA-4
3	16	90	11.11	2	44.44	ATA-5
4	16	60	16.67	2	66.67	ATA-5
5	16	40	25.00	2	100.00	ATA-6

ATA-4 UDMA Mode 2 is sometimes called Ultra-ATA/33 or ATA-33.

ATA-5 UDMA Mode 4 is sometimes called Ultra-ATA/66 or ATA-66.

ATA-6 UDMA Mode 5 is sometimes called Ultra-ATA/100 or ATA-100.

ATAPI (ATA Packet Interface)

ATAPI is a standard designed to provide the commands needed for devices such as CD-ROMs and tape drives that plug into an ordinary ATA (IDE) connector. The principal advantage of ATAPI hardware is that it's cheap and works on your current adapter. For CD-ROMs, it has a somewhat lower CPU usage compared to proprietary adapters, but there's no performance gain otherwise. For tape drives, ATAPI has potential for superior performance and reliability compared to the popular floppy controller attached tape devices. ATAPI is also used to run other removable storage devices, such as the LS-120 superdisk drives and internal Iomega Zip and Jaz drives.

Although ATAPI CD-ROMs use the hard disk interface, this does not mean they look like ordinary hard disks; to the contrary, from a software point of view, they are a completely different kind of animal. They actually most closely resemble a SCSI device. All modern IDE CD-ROMs support the ATAPI protocols, and generally the terms are synonymous. In other words, an ATAPI CD-ROM is an IDE CD-ROM and vice versa.

Caution

ATAPI support is not found directly in the BIOS of many systems. Systems without ATAPI support in the BIOS cannot boot from an ATAPI CD-ROM, and you still must load a driver to use ATAPI under DOS or Windows. Windows 95 and later (including 98 and Me) and Windows NT (including Windows 2000 and XP) have native ATAPI support, and newer systems with ATAPI-aware BIOSes are now available, which allow booting from an ATAPI CD-ROM. Some versions of the Windows 98, NT, 2000, and XP CD-ROMs are directly bootable on those systems, greatly easing installation.

I normally recommend keeping different types of IDE devices on separate channels. Some older chipsets cannot support setting different transfer rates for different devices, which means the channel must be set to the speed of the slowest device. Because most CD-ROM and tape drives run at lower IDE mode speeds, this forces your hard disk to run more slowly if they share a single cable. Even if the chipset you have supports separate speed settings for devices on the same channel (cable), I still recommend keeping them separate because IDE does not normally support overlapping access, such as SCSI. So, when one drive is running, the other cannot be accessed. By keeping the CD-ROM and hard disk on separate channels, you can more effectively overlap accessing between them.

Serial ATA

With the introduction of ATA-6, it seems that the parallel ATA standard that has been in use for more than 10 years is running out of steam. Sending data at rates faster than 100MB/sec down a parallel ribbon cable is fraught with all kinds of problems because of signal timing, electromagnetic interference (EMI), and other integrity problems. The solution is in a new ATA interface called serial ATA, which is an evolutionary backward-compatible replacement for the parallel ATA physical storage interface. Serial ATA is backward compatible in that it is compatible with existing software, which will run on the new architecture without any changes. In other words, the existing BIOS, operating systems, and utilities that work on parallel ATA will also work on serial ATA. This means Serial ATA supports all existing ATA and ATAPI devices, including CD-ROM and CD-RW drives, DVD drives, tape devices, SuperDisk drives, and any other storage device currently supported by parallel ATA.

Of course, they do differ physically—that is, you won't be able to plug parallel ATA drives into serial ATA host adapters and vice versa. The physical changes are all for the better because serial ATA uses much thinner cables with only 7 pins that are easier to route inside the PC and easier to plug in with smaller redesigned cable connectors. The interface chip designs also are improved with fewer pins and lower voltages. These improvements are all designed to eliminate the design problems inherent in parallel ATA.

Serial ATA (SATA) won't be integrated into PCs overnight; however, it is clear to me that it will eventually replace parallel ATA as the de facto standard internal storage device interface found in PCs. The transition from ATA to SATA is a gradual one, and during this transition parallel ATA capabilities will continue to be available. I would also expect that with more than a 10-year history, parallel ATA devices will continue to be available even after most PCs have gone to SATA.

Development for Serial ATA started when the Serial ATA Working Group effort was announced at the Intel Developer Forum in February 2000. The initial members of the Serial ATA Working Group included APT Technologies, Dell, IBM, Intel, Maxtor, Quantum, and Seagate. The first serial ATA specification 1.0 was completed in November 2000, and can be downloaded from the Serial ATA Working Group Web site at `http://www.serialata.org`. Since forming, the group has added more than 60 Contributor and Adopter companies to the membership from all areas of industry. It is expected that by the beginning of 2002, serial ATA will be shipping in many new systems.

The performance of SATA is impressive. Currently, three versions of the standard are available, which all use the same cables and connectors; they differ only in transfer rate performance. Initially, only the first version will be available, but the roadmap to doubling and quadrupling performance from there has been clearly established. Table 7.15 shows the specifications for each of the three SATA versions.

Table 7.16 SATA Standards Specifications

Serial ATA Type	Bus Width (bits)	Bus Speed (MHz)	Data Cycles per Clock	Bandwidth (MB/sec)
SATA-150	1	1500	1	150
SATA-300	1	3000	1	300
SATA-600	1	6000	1	600

From the table, you can see that serial ATA sends data only a single bit at a time. The cable used has only 7 wires and is a very thin design, with keyed connectors only 14mm (0.55 inches) wide on each end. This eliminates problems with airflow around the wider parallel ATA ribbon cables. Each cable has connectors only at each end and connects the device directly to the host adapter (normally on the motherboard). There are no master/slave settings because each cable supports only a single device. The cable ends are interchangeable—the connector on the motherboard is the same as on the device, and both cable ends are identical. Maximum SATA cable length is 1 meter (39.37 inches), which is considerably longer than the 18-inch maximum for parallel ATA. Even with this thinner, longer, and less expensive cable, transfer rates initially of 150MB/sec (one and a half times greater than parallel ATA/100) and in the future up to 300MB/sec and even 600MB/sec are possible.

Serial ATA uses a special encoding scheme called 8B/10B to encode and decode data sent along the cable. The 8B/10B transmission code originally was developed (and patented) by IBM in the early 1980s for use in high-speed data communications. This encoding scheme is now used by many high-speed data transmission standards, including Gigabit Ethernet, Fibre Channel, FireWire, and others. The main purpose of the 8B/10B encoding scheme is to guarantee that there are never more than four 0s (or 1s) transmitted consecutively. This is a form of Run Length Limited (RLL) encoding called RLL 0,4 in which the 0 represents the minimum and the 4 represents the maximum number of consecutive 0s in each encoded character.

8B/10B encoding also ensures that there are never more than six or less than four 0s (or 1s) in a single encoded 10-bit character. Because 1s and 0s are sent as voltage changes on a wire, this ensures that the spacing between the voltage transitions sent by the transmitter will be fairly balanced, with a more regular and steady stream of pulses. This presents a more steady load on the circuits, increasing reliability. The conversion from 8-bit data to 10-bit encoded characters for transmission leaves a number of 10-bit patterns unused. Several of these additional patterns are used to provide flow control, delimit packets of data, perform error checking, or perform other special needs.

The physical transmission scheme for SATA uses what is called differential NRZ (Non-Return to Zero). This uses a balanced pair of wires, each carrying plus or minus 0.25V (one quarter-volt). The signals are sent differentially: If one wire in the pair is carrying +0.25V, the other wire is carrying -0.25V, where the differential voltage between the two wires is always 0.5V (a half-volt). This means that for a given voltage waveform, the opposite voltage waveform is sent along the adjacent wire. Differential transmission minimizes electromagnetic radiation and makes the signals easier to read on the receiving end.

A 15-pin power cable and power connector is optional with SATA, providing 3.3V power in addition to the 5V and 12V provided via the industry-standard 4-pin device power connectors. Although it has 15 pins, this new power connector design is only 24mm (0.945 inches). With 3 pins designated for each of the 3.3V, 5V, and 12V power levels, enough capacity exists for up to 4.5 amps of current at each voltage, which is ample for even the most power-hungry drives. For compatibility with existing power supplies, SATA drives can be made with either the original, standard 4-pin device power connector or the new 15-pin SATA power connector, or both.

Figure 7.6 shows what the new SATA signal and power connectors look like.

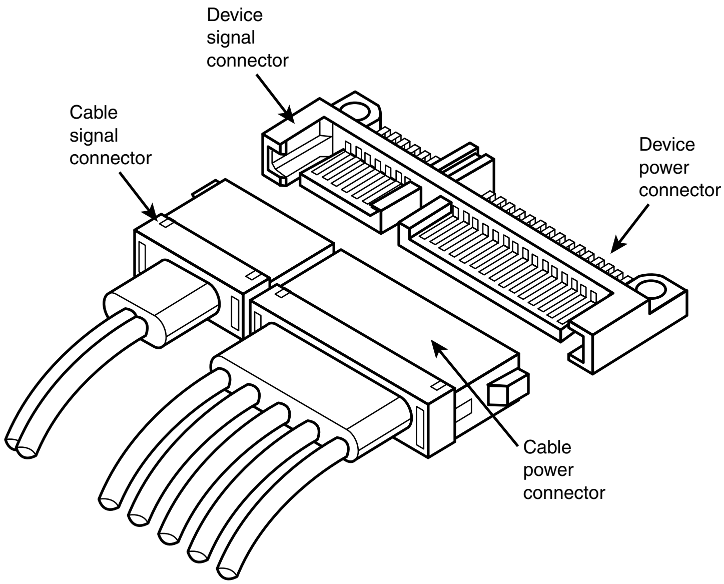

Figure 7.6 SATA (Serial ATA) signal and power connectors.

The pinouts for the Serial ATA data and optional power connectors are shown in Tables 7.17 and 7.18.

Table 7.17 Serial ATA (SATA) Data Connector Pinout

Signal Pin	Signal	Description
S1	Gnd	First mate
S2	A+	Host Transmit +
S3	A-	Host Transmit -
S4	Gnd	First mate
S5	B-	Host Receive -
S6	B+	Host Receive +
S7	Gnd	First mate

All pins are in a single row spaced 1.27mm (.050") apart.
All ground pins are longer so they will make contact before the signal/power pins to allow hot plugging.

Table 7.18 Serial ATA (SATA) Optional Power Connector Pinout

Power Pin	Signal	Description
P1	+3.3V	3.3V power
P2	+3.3V	3.3V power
P3	+3.3V	3.3V power

Table 7.18 Continued

Power Pin	Signal	Description
P4	Gnd	First mate
P5	Gnd	First mate
P6	Gnd	First mate
P7	+5V	5V power
P8	+5V	5V power
P9	+5V	5V power
P10	Gnd	First mate
P11	Gnd	First mate
P12	Gnd	First mate
P13	+12V	12V power
P14	+12V	12V power
P15	+12V	12V power

All pins are in a single row spaced 1.27mm (.050") apart.

All ground pins are longer so they will make contact before the signal/power pins to allow hot plugging.

Three power pins are used to carry 4.5 A maximum current for each voltage.

Configuration of serial ATA devices is also much simpler because the master/slave or cable select jumper settings used with parallel ATA are no longer necessary.

As with parallel ATA, serial ATA was designed to be the primary storage interface used inside a PC and was not designed to be used as an external interface. As such, SATA is not designed to compete with high-speed external device interfaces, such as SCSI, USB 2.0, or IEEE-1394 (FireWire). Therefore, I expect it to replace parallel ATA in systems over the next few years.

ATA RAID

RAID is an acronym for Redundant Array of Independent (or Inexpensive) Disks and was designed to improve the fault tolerance and performance of computer storage systems. RAID was first developed at the University of California at Berkeley in 1987, and was designed so that a group of smaller, less expensive drives could be interconnected with special hardware and software to make them appear as a single larger drive to the system. By using multiple drives to act as one drive, increases in fault tolerance and performance could be realized.

Initially, RAID was conceived to simply have all the individual drives in the array work together as a single larger drive with the combined storage space of all the individual drives added up. However, this actually reduced reliability and didn't do much for performance, either. For example, if you had four drives connected in an array acting as one drive, you would be four times as likely to experience a drive failure than if you used just a single larger drive. To improve on the reliability and performance, the Berkeley scientists proposed six levels (corresponding to different methods) of RAID. These levels provide varying emphasis on either fault tolerance (reliability), storage capacity, performance, or a combination of the three.

An organization called the RAID Advisory Board (RAB) was formed in July of 1992 to standardize, classify, and educate on the subject of RAID. They can be reached on the Web at `http://www.raid-advisory.com`. The RAB has developed specifications for RAID, a conformance program for the various RAID levels, and a classification program for RAID hardware.

Currently, seven standard RAID levels are defined by the RAID Advisory Board, called RAID 0 through 6. RAID normally is implemented by a RAID controller board, although software-only implementations are possible (but not recommended). The levels are as follows:

- *RAID Level 0—Striping*. File data is written simultaneously to multiple drives in the array, which act as a single larger drive. Offers high read/write performance but very low reliability. Requires a minimum of two drives to implement.
- *RAID Level 1—Mirroring*. Data written to one drive is duplicated on another, providing excellent fault tolerance (if one drive fails, the other will be used and no data lost), but no real increase in performance as compared to a single drive. Requires a minimum of two drives to implement (same capacity as one drive).
- *RAID Level 2—Bit-level ECC*. Data is split one bit at a time across multiple drives, and Error Correction Codes are written to other drives. Intended for storage devices that do not incorporate ECC internally (all SCSI and ATA drives have internal ECC). Provides high data rates with good fault tolerance, but large numbers of drives are required, and no commercial RAID 2 controllers or drives without ECC are available on the market that I am aware of.
- *RAID Level 3—Striped with parity*. Combines RAID Level 0 striping with an additional drive used for parity information. This RAID level is really an adaptation of RAID Level 0 that sacrifices some capacity, for the same number of drives. However, it also achieves a high level of data integrity or fault tolerance because data usually can be rebuilt if one drive fails. Requires a minimum of three drives to implement (two or more for data and one for parity).
- *RAID Level 4—Blocked data with parity*. Similar to RAID 3 except data is written in larger blocks to the independent drives, offering faster read performance with larger files. Requires a minimum of three drives to implement (two or more for data and one for parity).
- *RAID Level 5—Blocked data with distributed parity*. Similar to RAID 4 but offers improved performance by distributing the parity stripes over a series of hard drives. Requires a minimum of three drives to implement (two or more for data and one for parity).
- *RAID Level 6—Blocked data with double distributed parity*. Similar to RAID 5 except parity information is written twice using two different parity schemes to provide even better fault tolerance in case of multiple drive failures. Requires a minimum of four drives to implement (two or more for data and two for parity).

Additional RAID levels exist that are not supported by the RAID Advisory Board but which are instead custom implementations by specific companies. Note that the higher the number doesn't necessarily mean increased performance or fault tolerance; the numbered order of the RAID levels was entirely arbitrary.

Until recently, virtually all RAID controllers were SCSI based, meaning they used SCSI drives. For a professional setup, SCSI RAID is definitely the way to go because it combines the advantages of RAID with the advantages of SCSI—an interface that already was designed to support multiple drives. Now, however, ATA RAID controllers are available, allowing for even less expensive RAID implementations. These ATA RAID controllers usually are used in single-user systems for performance rather than reliability increases.

Most ATA RAID implementations are much simpler than the professional SCSI RAID adapters used on network file servers. ATA RAID is designed more for the individual who is seeking performance or simple drive mirroring for redundancy. When set up for performance, ATA RAID adapters run RAID Level 0, which incorporates data striping. Unfortunately, RAID 0 also sacrifices reliability such that if one drive fails, all data is lost. With RAID 0, performance scales up with the number of drives you add in the array. If you use four drives, you won't necessarily have four times the performance of a single drive, but it can be close to that for sustained transfers. Some overhead is still involved in the controller performing the striping and issues still exist with latency—that is, how long it takes to find the data—but performance will be higher than any single drive can normally achieve.

When set up for reliability, ATA RAID adapters generally run RAID Level 1, which is simple drive mirroring. All data written to one drive is written to the other. If one drive fails, the system can continue to work on the other drive. Unfortunately, this does not increase performance at all, and it also means you get to use only half of the available drive capacity. In other words, you must install two drives, but you get to use only one (the other is the mirror).

Combining performance with fault tolerance requires using one of the other RAID levels, such as 3 or 5. For example, virtually all professional RAID controllers used in network file servers are designed to use RAID Level 5. Controllers that implement RAID Level 5 are more expensive, and at least three drives must be connected. To improve reliability, but at a lower cost, many of the ATA RAID controllers enable combinations of the RAID levels—such as 0 and 1 combined. This usually requires four drives, two of which are striped together in a RAID Level 0 arrangement, which is then redundantly written to a second set of two drives in a RAID Level 1 arrangement. This enables you to have approximately double the performance of a single drive, and you have a backup set should one of the primary set fail.

Today, you can get IDE RAID controllers from companies such as Arco Computer Products, Iwill, Promise Technology, and more. A typical example of an ATA RAID controller is the Promise FastTrak 100/TX4. This controller enables up to four drives to be attached, and you can run them in RAID Level 0, 1, or 0+1 mode. This card also uses separate ATA data channels (cables) for each drive, allowing maximum performance. Promise Technology also has a less expensive ATA RAID card called the FastTrak 100/TX2, which has only two channels. You can still attach up to four drives, but because they share two ATA cables, performance isn't as good as if they were on separate cables. This is because only one drive can transfer on the cable at a time, which cuts performance in half.

If you are looking for an ATA RAID controller, things to look for include

- RAID levels supported (most support 0, 1, and 0+1 combined)
- Two or four channels
- Support for ATA/100 speeds
- Support for 33MHz or 66MHz PCI slots

If you want to experiment with RAID inexpensively, you can implement RAID without a custom controller when using certain higher-end (often server-based) operating systems. For example, Windows NT/2000 and XP Server operating systems provide a software implementation for RAID using both striping and mirroring. In these operating systems, the Disk Administrator tool is used to set up and control the RAID functions, as well as to reconstruct the volume when a failure has occurred. Normally, though, if you are building a server in which the ultimate in performance and reliability are desired, you should look for ATA or SCSI RAID controllers that support RAID Level 3 or 5.