

Hard Drive Technologies

In this chapter, you will learn how to

- Explain how hard drives work
- Identify and explain the ATA hard drive interfaces
- Identify and explain the SCSI hard drive interfaces
- Describe how to protect data with RAID
- Explain how to install drives
- Configure CMOS and install drivers
- Troubleshoot hard drive installation

Of all the hardware on a PC, none gets more attention—or gives more anguish—than the hard drive. There’s a good reason for this: if the hard drive breaks, you lose data. As you probably know, when the data goes, you have to redo work or restore from backup—or worse. It’s good to worry about the data, because the data runs the office, maintains the payrolls, and stores the e-mail. This level of concern is so strong that even the most neophyte PC users are exposed to terms such as *IDE*, *ATA*, and *controller*—even if they don’t put the terms into practice!

This chapter focuses on how hard drives work, beginning with the internal layout and organization of the hard drive. You’ll look at the different types of hard drives used today (PATA, SATA, and SCSI), how they interface with the PC, and how to install them properly into a system. The chapter covers how more than one drive may work with other drives to provide data safety and improve speed through a feature called RAID. Let’s get started.



NOTE Chapter 10, “Implementing Hard Drives,” continues the hard drive discussion by adding in the operating systems, showing you how to prepare drives to receive data, and teaching you how to maintain and upgrade drives in all versions of Windows.

Historical/Conceptual

How Hard Drives Work

All hard drives are composed of individual disks, or *platters*, with read/write heads on actuator arms controlled by a servo motor—all contained in a sealed case that prevents contamination by outside air (see Figure 9-1).



Figure 9-1 Inside the hard drive

The aluminum platters are coated with a magnetic medium. Two tiny read/write heads service each platter, one to read the top and the other to read the bottom of the platter (see Figure 9-2).



Figure 9-2 Top and bottom read/write heads and armatures

The coating on the platters is phenomenally smooth! It has to be, as the read/write heads actually float on a cushion of air above the platters, which spin at speeds between 3,500 and 10,000 rpm. The distance (flying height) between the heads and the disk surface is less than the thickness of a fingerprint. The closer the read/write heads are to the platter, the more densely the data packs onto the drive. These infinitesimal tolerances demand that the platters never be exposed to outside air. Even a tiny dust particle on a platter would act like a mountain in the way of the read/write heads and would cause catastrophic damage to the drive. To keep the air clean inside the drive, all hard drives use a tiny, heavily filtered aperture to keep the air pressure equalized between the interior and the exterior of the drive.

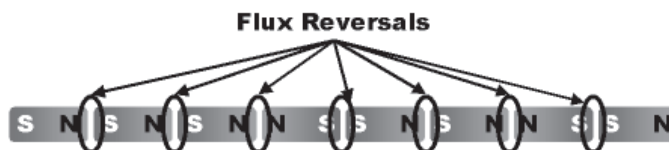
Data Encoding

Although the hard drive stores data in binary form, visualizing a magnetized spot representing a one and a non-magnetized spot representing a zero grossly oversimplifies the process. Hard drives store data in tiny magnetic fields—think of them as tiny magnets that can be placed in either direction on the platter, as shown in Figure 9-3. Each tiny magnetic field, called a *flux*, can switch back and forth through a process called a *flux reversal* (see Figure 9-4). Electronic equipment can read and write flux reversals much faster and easier than it can magnetize or not magnetize a spot to store a one or a zero.

Figure 9-3
Data is stored in
tiny magnetic fields.



Figure 9-4
Flux reversals



In early hard drives, as the read/write head moved over a spot, the direction of the flux reversal defined a one or a zero. As the read/write head passed from the left to the right, it recognized fluxes in one direction as a zero and the other direction as a one (Figure 9-5). Hard drives read these flux reversals at a very high speed when accessing or writing data.

Figure 9-5

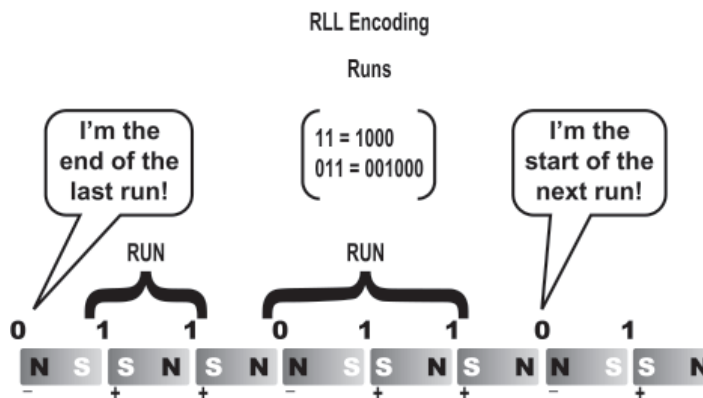
Fluxes are read in one direction as 0 and the other direction as 1.



Today's hard drives use a more complex and efficient method to interpret flux reversals using special data encoding systems. Instead of reading individual fluxes, a modern hard drive reads groups of fluxes called *runs*. Starting around 1991, hard drives began using a data encoding system known as *run length limited (RLL)*. With RLL, any combination of ones and zeroes can be stored in a preset combination of about 15 different runs. The hard drive looks for these runs and reads them as a group, resulting in much faster and much more dense data. Whenever you see RLL, you also see two numbers: the minimum and the maximum run length, such as RLL 1,7 or RLL 2,7. Figure 9-6 shows two sequential RLL runs.

Figure 9-6

Sequential RLL runs



Current drives use an extremely advanced method of RLL called *Partial Response Maximum Likelihood (PRML)* encoding. As hard drives pack more and more fluxes on the drive, the individual fluxes start to interact with each other, making it more and more difficult for the drive to verify where one flux stops and another starts. PRML uses powerful, intelligent circuitry to analyze each flux reversal and to make a “best guess” as to what type of flux reversal it just read. As a result, the maximum run length for PRML drives reaches up to around 16 to 20 fluxes, far more than the 7 or so on RLL drives. Longer run lengths enable the hard drive to use more complicated run combina-

tions so that the hard drive can store a phenomenal amount of data. For example, a run of only 12 fluxes on a hard drive might equal a string of 30 or 40 ones and zeroes when handed to the system from the hard drive.

The size required by each magnetic flux on a hard drive has reduced considerably over the years, resulting in higher capacities. As fluxes become smaller, they begin to interfere with each other in weird ways. I have to say *weird* since to make sense of what's going on at this subatomic level (I told you these fluxes were small!) would require that you take a semester of quantum mechanics! Let's just say that laying fluxes flat against the platter has reached its limit. To get around this problem, hard drive makers recently began to make hard drives that store their flux reversals vertically (up and down) rather than longitudinally (forward and backward), enabling them to make hard drives in the 1 terabyte (1,024 gigabyte) range. Manufacturers call this vertical storage method *perpendicular recording* (Figure 9-7).

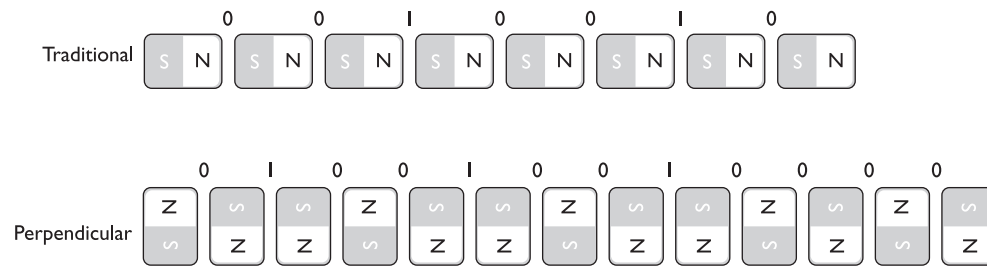


Figure 9-7 Perpendicular versus traditional longitudinal recording

For all this discussion and detail on data encoding, the day-to-day PC technician never deals with encoding. Sometimes, however, knowing what you don't need to know helps as much as knowing what you do need to know. Fortunately, data encoding is inherent to the hard drive and completely invisible to the system. You're never going to have to deal with data encoding, but you'll sure sound smart when talking to other PC techs if you know your RLL from your PRML!

Moving the Arms

The read/write heads move across the platter on the ends of *actuator arms*. In the entire history of hard drives, manufacturers have used only two technologies to move the arms: *stepper motor* and *voice coil*. Hard drives first used stepper motor technology, but today they've all moved to voice coil.

Stepper motor technology moved the arm in fixed increments or steps, but the technology had several limitations that doomed it. Because the interface between motor and actuator arm required minimal slippage to ensure precise and reproducible movements, the positioning of the arms became less precise over time. This physical deterioration caused data transfer errors. Additionally, heat deformation wreaked havoc with stepper motor drives. Just as valve clearances in automobile engines change with operating temperature, the positioning accuracy changed as the PC operated and various

hard drive components got warmer. Although very small, these changes caused problems. Accessing the data written on a cold hard drive, for example, became difficult after the disk warmed. In addition, the read/write heads could damage the disk surface if not “parked” (set in a non-data area) when not in use, requiring techs to use special parking programs before transporting a stepper motor drive.



NOTE Floppy disk drives still use stepper motors.

All hard drives made today employ a linear motor to move the actuator arms. The linear motor, more popularly called a voice coil motor, uses a permanent magnet surrounding a coil on the actuator arm. When an electrical current passes, the coil generates a magnetic field that moves the actuator arm. The direction of the actuator arm’s movement depends on the polarity of the electrical current through the coil. Because the voice coil and the actuator arm never touch, no degradation in positional accuracy takes place over time. Voice coil drives automatically park the heads when the drive loses power, making the old stepper motor park programs obsolete.

Lacking the discrete “steps” of the stepper motor drive, a voice coil drive cannot accurately predict the movement of the heads across the disk. To make sure voice coil drives land exactly in the correct area, the drive reserves one side of one platter for navigational purposes. This area essentially “maps” the exact location of the data on the drive. The voice coil moves the read/write head to its best guess about the correct position on the hard drive. The read/write head then uses this map to fine-tune its true position and make any necessary adjustments.

Now that you have a basic understanding of how a drive physically stores data, let’s turn to how the hard drive organizes that data so we can use that drive.

Geometry

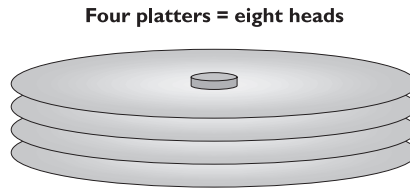
Have you ever seen a cassette tape? If you look at the actual brown Mylar (a type of plastic) tape, nothing will tell you whether sound is recorded on that tape. Assuming the tape is not blank, however, you know *something* is on the tape. Cassettes store music in distinct magnetized lines. You could say that the physical placement of those lines of magnetism is the tape’s “geometry.”

Geometry also determines where a hard drive stores data. As with a cassette tape, if you opened up a hard drive, you would not see the geometry. But rest assured that the drive has geometry; in fact, every model of hard drive uses a different geometry. We describe the geometry for a particular hard drive with a set of numbers representing three values: heads, cylinders, and sectors per track.

Heads

The number of *heads* for a specific hard drive describes, rather logically, the number of read/write heads used by the drive to store data. Every platter requires two heads. If a hard drive has four platters, for example, it would need eight heads (see Figure 9-8).

Figure 9-8
Two heads
per platter

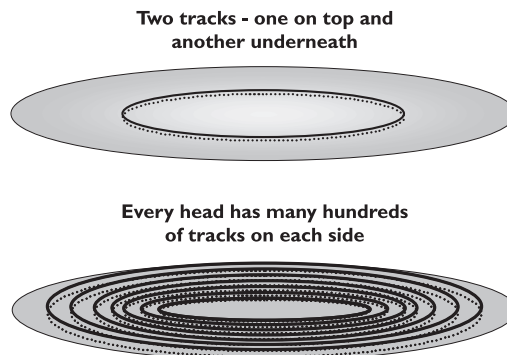


Based on this description of heads, you would think that hard drives would always have an even number of heads, right? Wrong! Most hard drives reserve a head or two for their own use. Therefore, a hard drive can have either an even or an odd number of heads.

Cylinders

To visualize cylinders, imagine taking a soup can and opening both ends of the can. Wash off the label and clean out the inside. Now look at the shape of the can; it is a geometric shape called a cylinder. Now imagine taking that cylinder and sharpening one end so that it easily cuts through the hardest metal. Visualize placing the ex-soup can over the hard drive and pushing it down through the drive. The can cuts into one side and out the other of each platter. Each circle transcribed by the can is where you store data on the drive, and is called a *track* (Figure 9-9).

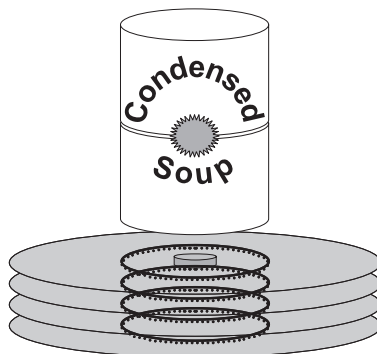
Figure 9-9
Tracks



Each side of each platter contains tens of thousands of tracks. Interestingly enough, the individual tracks themselves are not directly part of the drive geometry. Our interest

lies only in the groups of tracks of the same diameter, going all of the way through the drive. Each group of tracks of the same diameter is called a *cylinder* (see Figure 9-10).

Figure 9-10
Cylinder



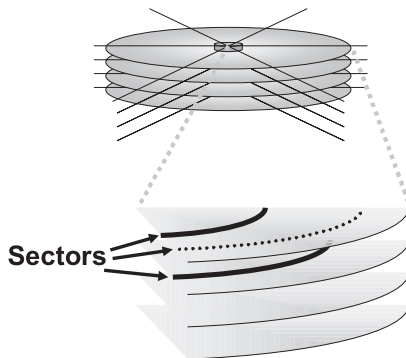
**All tracks of the same diameter
are called a cylinder**

There's more than one cylinder! Go get yourself about a thousand more cans, each one a different diameter, and push them through the hard drive. A typical hard drive contains thousands of cylinders.

Sectors per Track

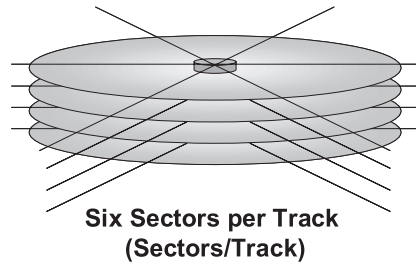
Now imagine cutting the hard drive like a birthday cake, slicing all the tracks into tens of thousands of small slivers. Each sliver is called a *sector*, and each sector stores 512 bytes of data (see Figure 9-11). Note that *sector* refers to the sliver when discussing geometry, but it refers to the specific spot on a single track within that sliver when discussing data capacity.

Figure 9-11
Sectors



The sector is the universal “atom” of all hard drives. You can't divide data into anything smaller than a sector. Although sectors are important, the number of sectors is not a geometry. The geometry value is called *sectors per track* (sectors/track). The sectors/track value describes the number of sectors in each track (see Figure 9-12).

Figure 9-12
Sectors per track



The Big Three

Cylinders, heads, and sectors/track combine to define the hard drive's geometry. In most cases, these three critical values are referred to as *CHS*. The importance of these three values lies in the fact that the PC's BIOS needs to know the drive's geometry to know how to talk to the drive. Back in the old days, a technician needed to enter these values into the CMOS setup program manually. Today, every hard drive stores the CHS information in the drive itself, in an electronic format that enables the BIOS to query the drive automatically to determine these values. You'll see more on this later in the chapter in the section called "Autodetection."

Two other values—write precompensation cylinder and landing zone—no longer have relevance in today's PCs; however, these terms still are tossed around and a few CMOS setup utilities still support them—another classic example of a technology appendix! Let's look at these two holdouts from another era so when you access CMOS, you won't say, "What the heck are these?"

Write Precompensation Cylinder

Older hard drives had a real problem with the fact that sectors toward the inside of the drives were much smaller than sectors toward the outside. To handle this, an older drive would spread data a little farther apart once it got to a particular cylinder. This cylinder was called the write precompensation (write precomp) cylinder, and the PC had to know which cylinder began this wider data spacing. Hard drives no longer have this problem, making the write precomp setting obsolete.

Landing Zone

On older hard drives with stepper motors, the landing zone value designated an unused cylinder as a "parking place" for the read/write heads. As mentioned earlier, old stepper motor hard drives needed to have their read/write heads parked before being moved in order to avoid accidental damage. Today's voice coil drives park themselves whenever they're not accessing data, automatically placing the read/write heads on the landing zone. As a result, the BIOS no longer needs the landing zone geometry.

Essentials

ATA—The King

Over the years, many interfaces existed for hard drives, with such names as ST-506 and ESDI. Don't worry about what these abbreviations stood for; neither the CompTIA A+

certification exams nor the computer world at large has an interest in these prehistoric interfaces. Starting around 1990, an interface called ATA appeared that now virtually monopolizes the hard drive market. Only one other type of interface, the moderately popular small computer system interface (SCSI), has any relevance for hard drives.



NOTE The term *IDE* (integrated drive electronics) refers to any hard drive with a built-in controller. All hard drives are technically IDE drives, although we only use the term IDE when discussing ATA drives.

ATA drives come in two basic flavors: parallel ATA (PATA) and serial ATA (SATA). PATA drives dominated the industry for more than a decade but are being replaced by SATA drives. The leap from PATA to SATA is only one of a large number of changes that have taken place over the years with ATA. To appreciate these changes, we'll run through the many ATA standards forwarded over the years.



NOTE Modern external drives connect to a FireWire, Hi-Speed USB 2.0, or eSATA port. All three interfaces offer high data transfer rates and hot-swap capability, making them ideal for transporting huge files such as digital video clips. Regardless of the external interface, however, inside the casing you'll find an ordinary PATA or SATA drive, just like those described in this chapter.

ATA-I

When IBM unveiled the 80286-powered IBM PC AT in the early 1980s, it introduced the first PC to include BIOS support for hard drives. This BIOS supported up to two physical drives, and each drive could be up to 504 MB—far larger than the 5-MB and 10-MB drives of the time. Although having built-in support for hard drives certainly improved the power of the PC, at that time, installing, configuring, and troubleshooting hard drives could at best be called difficult.

To address these problems, Western Digital and Compaq developed a new hard drive interface and placed this specification before the *American National Standards Institute* (ANSI) committees, which in turn put out the *AT Attachment* (ATA) interface in March of 1989. The ATA interface specified a cable and a built-in controller on the drive itself. Most importantly, the ATA standard used the existing AT BIOS on a PC, which meant that you didn't have to replace the old system BIOS to make the drive work—a very important consideration for compatibility but one that would later haunt ATA drives. The official name for the standard, ATA, never made it into the common vernacular until recently, and then only as PATA to distinguish it from SATA drives.



NOTE The ANSI subcommittee directly responsible for the ATA standard is called Technical Committee T13. If you want to know what's happening with ATA, check out the T13 Web site: www.t13.org.

Early ATA Physical Connections

The first ATA drives connected to the computer via a 40-pin plug and a controller. Figure 9-13 shows the “business end” of an early ATA drive, with the connectors for the controller and the power cable.



Figure 9-13 Back of IDE drive showing 40-pin connector (left), jumpers (center), and power connector (right)

The controller is the support circuitry that acts as the intermediary between the hard drive and the external data bus. Electronically, the setup looks like Figure 9-14.

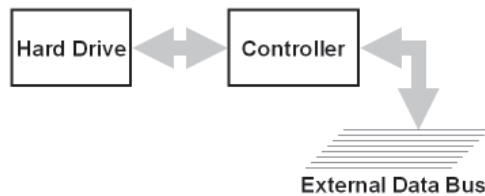


Figure 9-14 Relation of drive, controller, and bus

Wait a minute! If ATA drives are IDE, they already have a built-in controller. Why do they then have to plug into a *controller* on the motherboard? Well, this is a great example of a term that’s not used properly, but everyone (including the motherboard and hard drive makers) uses it this way. What we call the ATA controller is really no more than an interface providing a connection to the rest of the PC system. When your BIOS talks to the hard drive, it actually talks to the onboard circuitry on the drive, not the connection on the motherboard. But, even though the *real* controller resides on the hard drive, the 40-pin connection on the motherboard is called the controller. We have a lot of misnomers to live with in the ATA world!

The ATA-1 standard defined that no more than two drives attach to a single IDE connector on a single ribbon cable. Because up to two drives can attach to one connector via a single cable, you need to be able to identify each drive on the cable. The ATA

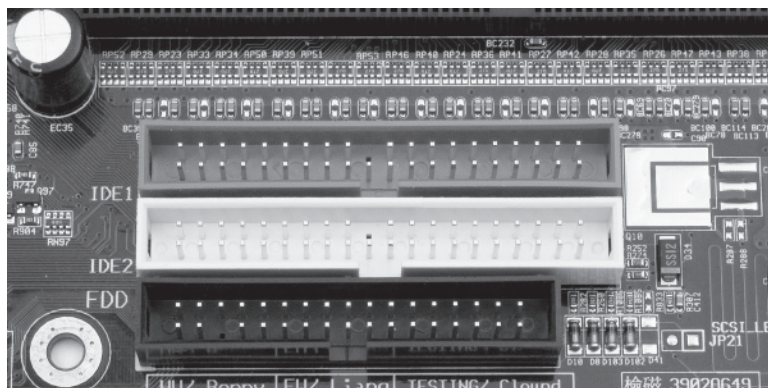
standard identifies the two different drives as “master” and “slave.” You set one drive as master and one as slave using tiny jumpers on the drives (Figure 9-15).



Figure 9-15 A typical hard drive with directions (top) for setting a jumper (bottom)

The controllers are on the motherboard and manifest themselves as two 40-pin male ports, as shown in Figure 9-16.

Figure 9-16
IDE interfaces on
a motherboard



PIO and DMA Modes

If you’re making a hard drive standard, you must define both the method and the speed at which the data’s going to move. ATA-1 defined two different methods, the first using programmed I/O (PIO) addressing and the second using direct memory access (DMA) mode.

PIO is nothing more than the traditional I/O addressing scheme, where the CPU talks directly to the hard drive via the BIOS to send and receive data. Three different PIO speeds called *PIO modes* were initially adopted:

- PIO mode 0: 3.3 MBps (megabytes per second)
- PIO mode 1: 5.2 MBps
- PIO mode 2: 8.3 MBps

DMA modes defined a method to enable the hard drives to talk to RAM directly using old-style DMA commands. (The ATA folks called this *single word DMA*.) This old-style DMA was slow, and the resulting three ATA single word DMA modes were also slow:

- Single word DMA mode 0: 2.1 MBps
- Single word DMA mode 1: 4.2 MBps
- Single word DMA mode 2: 8.3 MBps

When a computer booted up, the BIOS queried the hard drive to see what modes it could use and would then automatically adjust to the fastest mode.

ATA-2

In 1990, the industry adopted a series of improvements to the ATA standard called ATA-2. Many people called these new features *Enhanced IDE (EIDE)*. EIDE was really no more than a marketing term invented by Western Digital, but it caught on in common vernacular and is still used today, although its use is fading. Regular IDE drives quickly disappeared, and by 1995, EIDE drives dominated the PC world. Figure 9-17 shows a typical EIDE drive.



Figure 9-17 EIDE drive



NOTE The terms *ATA*, *IDE*, and *EIDE* are used interchangeably.

ATA-2 was the most important ATA standard, as it included powerful new features such as higher capacities; support for non-hard drive storage devices; support for two more ATA devices, for a maximum of four; and substantially improved throughput.

Higher Capacity with LBA

IBM created the AT BIOS to support hard drives many years before IDE drives were invented, and every system had that BIOS. The developers of IDE made certain that the new drives would run from the same AT BIOS command set. With this capability, you could use the same CMOS and BIOS routines to talk to a much more advanced drive. Your motherboard or hard drive controller wouldn't become instantly obsolete when you installed a new hard drive.



NOTE Hard drive makers talk about hard drive capacities in millions and billions of bytes, not megabytes and gigabytes!

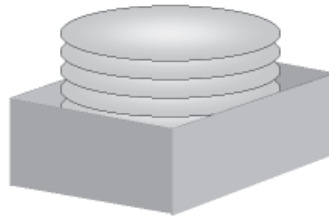
Unfortunately, the BIOS routines for the original AT command set allowed a hard drive size of only up to 528 million bytes (or 504 MB—remember that a mega = 1,048,576, not 1,000,000). A drive could have no more than 1,024 cylinders, 16 heads, and 63 sectors/track:

$$1,024 \text{ cylinders} \times 16 \text{ heads} \times 63 \text{ sectors/track} \times 512 \text{ bytes/sector} = 504 \text{ MB}$$

For years, this was not a problem. But when hard drives began to approach the 504 MB barrier, it became clear that there needed to be a way of getting past 504 MB. The ATA-2 standard defined a way to get past this limit with *logical block addressing (LBA)*. With LBA, the hard drive lies to the computer about its geometry through an advanced type of sector translation. Let's take a moment to understand sector translation, and then come back to LBA.

Sector Translation Long before hard drives approached the 504 MB limit, the limits of 1,024 cylinders, 16 heads, and 63 sectors/track gave hard drive makers fits. The big problem was the heads. Remember that every two heads means another platter, another physical disk that you have to squeeze into a hard drive. If you wanted a hard drive with the maximum number of 16 heads, you would need a hard drive with eight physical platters inside the drive! Nobody wanted that many platters: it made the drives too tall, it took more power to spin up the drive, and that many parts cost too much money (see Figure 9-18).

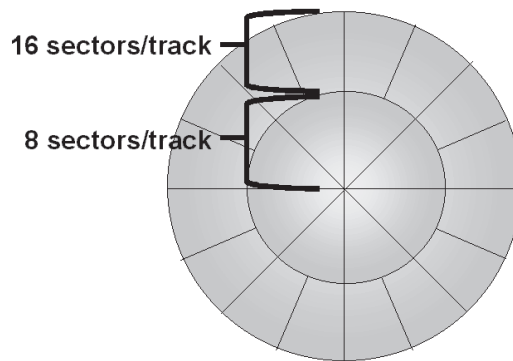
Figure 9-18
Too many heads



Looks a little high!

Manufacturers could readily produce a hard drive that had fewer heads and more cylinders, but the stupid 1,024/16/63 limit got in the way. Plus, the traditional sector arrangement wasted a lot of useful space. Sectors toward the inside of the drive, for example, are much shorter than the sectors on the outside. The sectors on the outside don't need to be that long, but with the traditional geometry setup, hard drive makers had no choice. They could make a hard drive store a lot more information, however, if hard drives could be made with more sectors/track on the outside tracks (see Figure 9-19).

Figure 9-19
Multiple sectors/
track



The ATA specification was designed to have two geometries. The *physical geometry* defined the real layout of the CHS inside the drive. The *logical geometry* described what the drive told the CMOS. In other words, the IDE drive “lied” to the CMOS, thus side-stepping the artificial limits of the BIOS. When data was being transferred to and from the drive, the onboard circuitry of the drive translated the logical geometry into the physical geometry. This function was, and still is, called *sector translation*.

Let's look at a couple of hypothetical examples in action. First, pretend that Seagate came out with a new, cheap, fast hard drive called the ST108. To get the ST108 drive fast and cheap, however, Seagate had to use a rather strange geometry, shown in Table 9-1.

ST108 Physical		BIOS Limits	
Cylinders	2,048	Cylinders	1,024
Heads	2	Heads	16
Sectors/Track	52	Sectors/Track	63
Total Capacity	108 MB		

Table 9-1 Seagate's ST108 Drive Geometry

Notice that the cylinder number is greater than 1,024. To overcome this problem, the IDE drive performs a sector translation that reports a geometry to the BIOS that is totally different from the true geometry of the drive. Table 9-2 shows the actual geometry and the “logical” geometry of our mythical ST108 drive. Notice that the logical geometry is now within the acceptable parameters of the BIOS limitations. Sector translation never changes the capacity of the drive; it changes only the geometry to stay within the BIOS limits.

Physical		Logical	
Cylinders	2,048	Cylinders	512
Heads	2	Heads	8
Sectors/Track	52	Sectors/Track	52
Total Capacity	108 MB	Total Capacity	108 MB

Table 9-2 Physical and Logical Geometry of the ST108 Drive

Back to LBA Now let's watch how the advanced sector translation of LBA provides support for hard drives greater than 504 MB. Let's use an old drive, the Western Digital WD2160, a 2.1-GB hard drive, as an example. This drive is no longer in production but its smaller CHS values make understanding LBA easier. Table 9-3 lists its physical and logical geometries.

Physical		Logical	
Cylinders	16,384	Cylinders	1,024
Heads	4	Heads	64
Sectors/Track	63	Sectors/Track	63
Total Capacity	2.1 GB	Total Capacity	2.1 GB

Table 9-3 Western Digital WD2160's Physical and Logical Geometries

Note that, even with sector translation, the number of heads is greater than the allowed 16! So here's where the magic of LBA comes in. The WD2160 is capable of LBA. Now assuming that the BIOS is also capable of LBA, here's what happens. When the computer boots up, the BIOS asks the drives if they can perform LBA. If they say yes, the BIOS and the drive work together to change the way they talk to each other. They can do this without conflicting with the original AT BIOS commands by taking advantage of unused commands to use up to 256 heads. LBA enables support for a maximum of $1,024 \times 256 \times 63 \times 512$ bytes = 8.4-GB hard drives. Back in 1990, 8.4 GB was hundreds of times larger than the drives used at the time. Don't worry, later ATA standards will get the BIOS up to today's huge drives!

Not Just Hard Drives Anymore:ATAPI

ATA-2 added an extension to the ATA specification, called Advanced Technology Attachment Packet Interface (ATAPI), that enabled non-hard drive devices such as CD-ROM drives and tape backups to connect to the PC via the ATA controllers. ATAPI drives have the same 40-pin interface and master/slave jumpers as ATA hard drives. Figure 9-20 shows an ATAPI CD-RW drive attached to a motherboard. The key difference between hard drives and every other type of drive that attaches to the ATA controller is in how the drives get BIOS support. Hard drives get it through the system BIOS, whereas non-hard drives require the operating system to load a software driver.

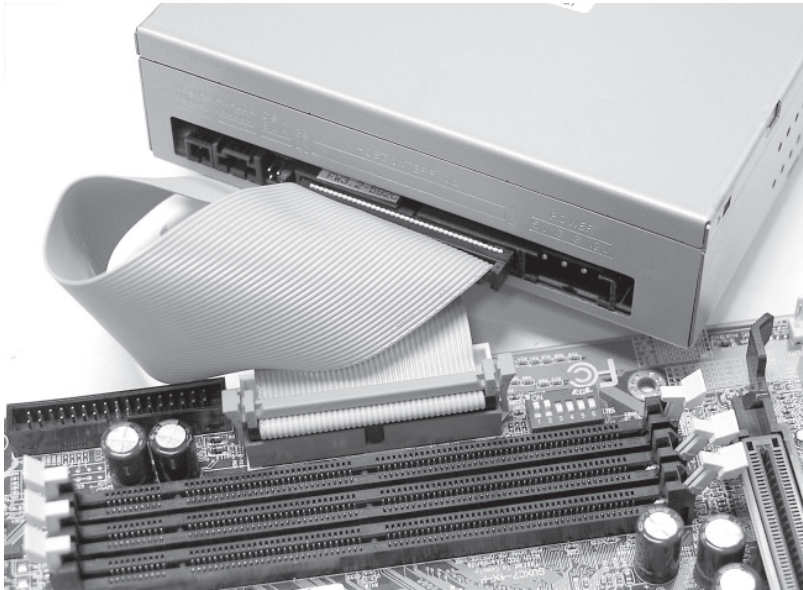


Figure 9-20 ATAPI CD-RW drive attached to a motherboard via a standard, 40-pin ribbon cable



NOTE With the introduction of ATAPI, the ATA standards are often referred to as ATA/ATAPI instead of just ATA.

More Drives with ATA-2

ATA-2 added support for a second controller, raising the total number of supported drives from two to four. Each of the two controllers is equal in power and capability. Figure 9-21 is a close-up of a typical motherboard, showing the primary controller marked as *IDE1* and the secondary marked as *IDE2*.

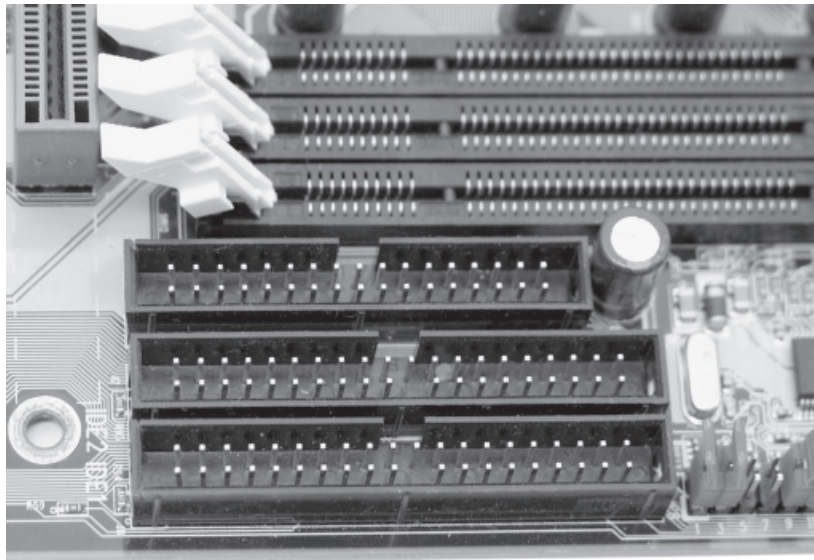


Figure 9-21 Primary and secondary controllers labeled on a motherboard

Increased Speed

ATA-2 defined two new PIO modes and a new type of DMA called *multi-word DMA* that was a substantial improvement over the old DMA. Technically, multi-word DMA was still the old-style DMA, but it worked in a much more efficient manner so it was much faster.

- PIO mode 3: 11.1 MBps
- PIO mode 4: 16.6 MBps
- Multi-word DMA mode 0: 4.2 MBps

- Multi-word DMA mode 1: 13.3 MBps
- Multi-word DMA mode 2: 16.6 MBps

ATA-3

ATA-3 came on quickly after ATA-2 and added one new feature called *Self-Monitoring, Analysis, and Reporting Technology* (S.M.A.R.T., one of the few PC acronyms that requires the use of periods after each letter). S.M.A.R.T. helps predict when a hard drive is going to fail by monitoring the hard drive’s mechanical components.

S.M.A.R.T. is a great idea and is popular in specialized server systems, but it’s complex, imperfect, and hard to understand. As a result, only a few utilities can read the S.M.A.R.T. data on your hard drive. Your best sources are the hard drive manufacturers. Every hard drive maker has a free diagnostic tool (that usually works only for their drives) that will do a S.M.A.R.T. check along with other tests. Figure 9-22 shows Western Digital’s Data Lifeguard Tool in action. Note that it says only whether the drive has passed or not. Figure 9-23 shows some S.M.A.R.T. information.

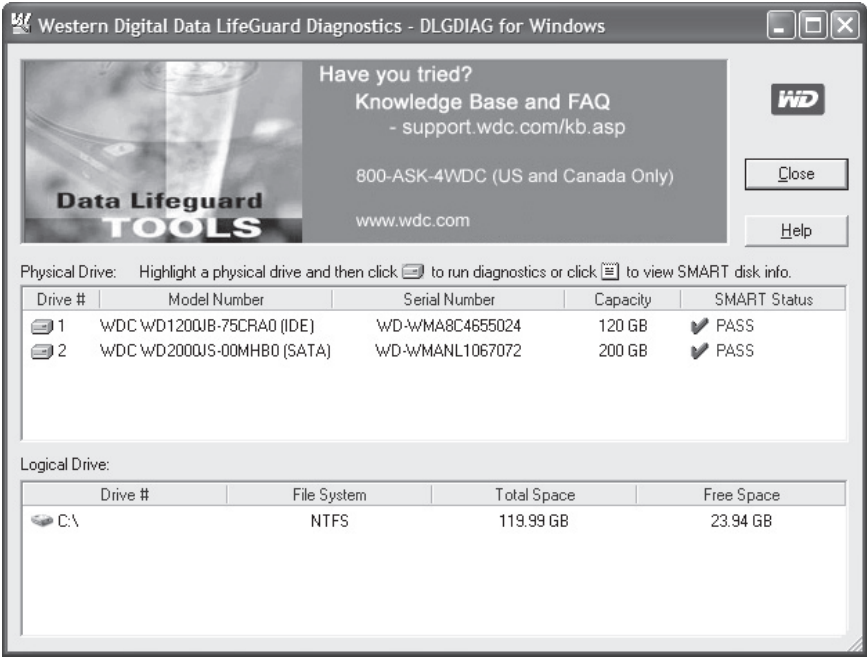


Figure 9-22 Data Lifeguard

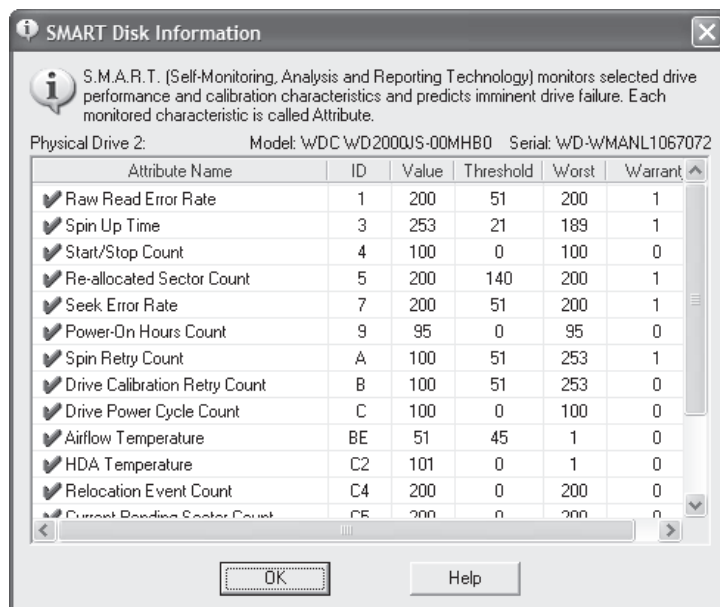


Figure 9-23 S.M.A.R.T. information

Although you can see the actual S.M.A.R.T. data, it's generally useless or indecipherable. It's best to trust the manufacturer's opinion and run the software provided.

ATA-4

Anyone who has opened a big database file on a hard drive appreciates that a faster hard drive is better. ATA-4 introduced a new DMA mode called Ultra-DMA that is now the primary way a hard drive communicates with a PC. Ultra DMA uses DMA bus mastering to achieve far faster speeds than was possible with PIO or old-style DMA. ATA-4 defined three Ultra DMA modes:

- Ultra DMA mode 0: 16.7 MBps
- Ultra DMA mode 1: 25.0 MBps
- Ultra DMA mode 2: 33.3 MBps



NOTE Ultra DMA mode 2, the most popular of the ATA-4 DMA modes, is also called ATA/33.

INT13 Extensions

Here's an interesting factoid for you: The original ATA-1 standard allowed for hard drives up to 137 GB! It wasn't the ATA standard that caused the 504-MB size limit, it was the fact that the standard used the old AT BIOS and the BIOS, not the ATA standard,

could support only 504 MB. LBA was a work-around that told the hard drive to lie to the BIOS to get it up to 8.4 GB. But eventually hard drives started edging close to the LBA limit and something had to be done. The T13 folks said, “This isn’t *our* problem! It’s the ancient BIOS problem. You BIOS makers need to fix the BIOS!” And they did.

In 1994, Phoenix Technologies (the BIOS manufacturer) came up with a new set of BIOS commands called *Interrupt 13 (INT13) extensions*. INT13 extensions broke the 8.4-GB barrier by completely ignoring the CHS values and instead feeding the LBA a stream of addressable sectors. A system with INT13 extensions can handle drives up to 137 GB. The entire PC industry quickly adopted INT13 extensions and every system made since 2000–2001 supports INT13 extensions.

ATA-5

Ultra DMA was such a huge hit that the ATA folks adopted two faster Ultra DMA modes with ATA-5:

- Ultra DMA mode 3: 44.4 MBps
- Ultra DMA mode 4: 66.6 MBps



NOTE Ultra DMA mode 4, the most popular of the ATA-5 DMA modes, is also called ATA/66.

Ultra DMA modes 4 and 5 ran so quickly that the ATA-5 standard defined a new type of ribbon cable of handling the higher speeds. This cable still has 40 pins on the connectors, but it includes another 40 wires in the cable that act as grounds to improve the cable’s ability to handle high-speed signals. Previous versions of ATA didn’t define where the different drives were plugged into the ribbon cable, but ATA-5 defined exactly where the controller, master, and slave drives connected, even defining colors to identify them. Take a look at the ATA/66 cable in Figure 9-24. The connector on the left is colored blue (which you could see if the photo was in color!)—that connector must be the one used to plug into the controller. The connector in the middle is grey—that’s for the slave drive. The connector on the right is black—that’s for the master drive. Any ATA/66 controller connections are colored blue to let you know it is an ATA/66 controller.

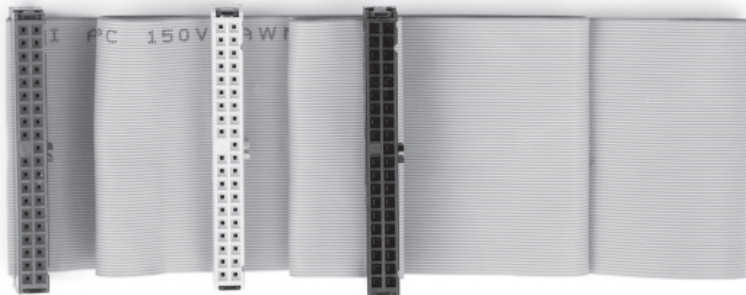


Figure 9-24 ATA/66 cable

ATA/66 is backward compatible, so you may safely plug an earlier drive into an ATA/66 cable and controller. If you plug an ATA/66 drive into an older controller it will work—just not in ATA/66 mode. The only risky action is to use an ATA/66 controller and hard drive with a non-ATA/66 cable. Doing so will almost certainly cause nasty data losses!

ATA-6

Hard drive size exploded in the early 21st century and the seemingly impossible-to-fill 137-GB limit created by INT13 extensions became a barrier to fine computing more quickly than most people had anticipated. When drives started hitting the 120-GB mark, the T13 committee adopted an industry proposal pushed by Maxtor (a major hard drive maker) called *Big Drive* that increased the limit to more than 144 petabytes (approximately 144,000,000 GB). T13 also thankfully gave the new standard a less-silly name, calling it ATA-6. Big Drive was basically just a 48-bit LBA, supplanting the older 24-bit addressing of LBA and INT13 extensions. Plus, the standard defined an enhanced block mode, enabling drives to transfer up to 65,536 sectors in one chunk, up from the measly 256 sectors of lesser drive technologies.

ATA-6 also introduced Ultra DMA mode 5, kicking the data transfer rate up to 100 MBps. Ultra DMA mode 5 is more commonly referred to as ATA/100, which requires the same 80-wire connectors as ATA/66.

ATA-7

ATA-7 brought two new innovations to the ATA world—one evolutionary and the other revolutionary. The evolutionary innovation came with the last of the parallel ATA Ultra DMA modes; the revolutionary was a new form of ATA called *serial ATA* (SATA).

ATA/133

ATA-7 introduced the fastest and probably least adopted of all the ATA speeds, Ultra DMA mode 6 (ATA/133). Even though it runs at a speed of 133 MBps, the fact that it came out with SATA kept many hard drive manufacturers away. ATA/133 uses the same cables as Ultra DMA 66 and 100.

While you won't find many ATA/133 hard drives, you will find plenty of ATA/133 controllers. There's a trend in the industry to color the controller connections on the hard drive red, although this is not part of the ATA-7 standard.

Serial ATA

The real story of ATA-7 is SATA. For all its longevity as the mass storage interface of choice for the PC, parallel ATA has problems. First, the flat ribbon cables impede air-flow and can be a pain to insert properly. Second, the cables have a limited length, only 18 inches. Third, you can't hot-swap PATA drives. You have to shut down completely before installing or replacing a drive. Finally, the technology has simply reached the limits of what it can do in terms of throughput.

Serial ATA addresses these issues. SATA creates a point-to-point connection between the SATA device—hard disk, CD-ROM, CD-RW, DVD-ROM, DVD-RW, and so on—and the SATA controller. At a glance, SATA devices look identical to standard PATA devices. Take a closer look at the cable and power connectors, however, and you'll see significant differences (Figure 9-25).

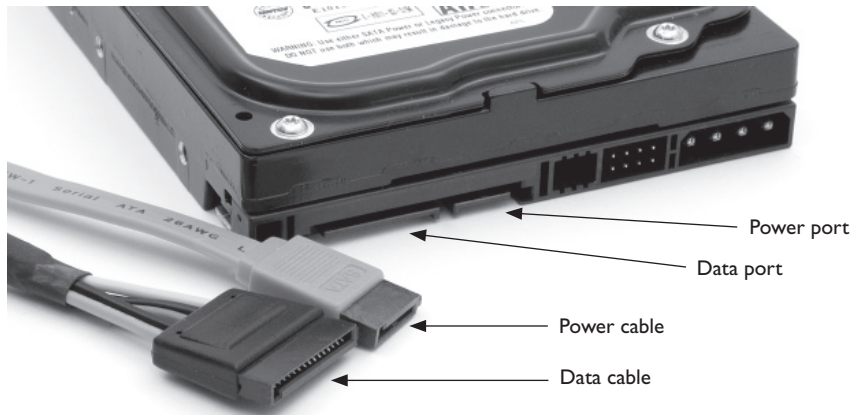


Figure 9-25 SATA hard drive cables and connectors

Because SATA devices send data serially instead of in parallel, the SATA interface needs far fewer physical wires—seven instead of the eighty wires that is typical of PATA—resulting in much thinner cabling. This might not seem significant, but the benefit is that thinner cabling means better cable control and better airflow through the PC case, resulting in better cooling.

Further, the maximum SATA device cable length is more than twice that of an IDE cable—about 40 inches (1 meter) instead of 18 inches. Again, this might not seem like a big deal, unless you've struggled to connect a PATA hard disk installed into the top bay of a full-tower case to an IDE connector located all the way at the bottom of the motherboard!

SATA does away with the entire master/slave concept. Each drive connects to one port, so no more daisy-chaining drives. Further, there's no maximum number of drives—many motherboards are now available that support up to eight SATA drives. Want more? Snap in a SATA host card and load 'em up!

The big news, however, is in data throughput. As the name implies, SATA devices transfer data in serial bursts instead of parallel, as PATA devices do. Typically, you might not think of serial devices as being faster than parallel, but in this case, that's exactly the case. A SATA device's single stream of data moves much faster than the multiple streams of data coming from a parallel IDE device—theoretically up to 30 times faster! SATA drives come in two common varieties, the 1.5Gb and the 3Gb, that have a maximum throughput of 150 MBps and 300 MBps, respectively.

Number-savvy readers might have noticed a discrepancy between the names and throughput of the two SATA drives. After all, 1.5 Gb per second throughput translates to 192 MB per second, a lot higher than the advertised speed of a “mere” 150 MBps. The same is true of the 3Gb/300 MBps drives. The encoding scheme used on SATA drives takes about 20 percent of the overhead for the drive, leaving 80 percent for pure bandwidth. The 3Gb drive created all kinds of problems, because the name of the committee working on the specifications was called the SATA II committee, and marketers picked up on the SATA II name. As a result, you’ll find many brands called SATA II rather than 3Gb. The SATA committee now goes by the name SATA-IO.

SATA is backward-compatible with current PATA standards and enables you to install a parallel ATA device, including a hard drive, CD-media drive, and other devices, to a serial ATA controller by using a SATA bridge. A SATA bridge manifests as a tiny card that you plug directly into the 40-pin connector on a PATA drive. As you can see in Figure 9-26, the controller chip on the bridge requires separate power; you plug a Molex connector into the PATA drive as normal. When you boot the system, the PATA drive shows up to the system as a SATA drive.



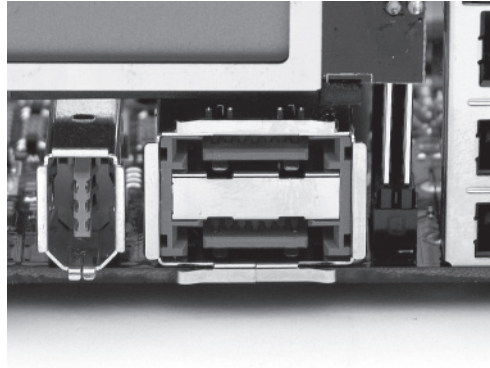
Figure 9-26 SATA bridge

SATA's ease of use has made it the choice for desktop system storage, and its success is already showing in the fact that more than 90 percent of all hard drives sold today are SATA drives.

eSATA

External SATA (eSATA) extends the SATA bus to external devices, as the name would imply. The eSATA drives use similar connectors to internal SATA, but they're keyed differently so you can't mistake one for the other. Figure 9-27 shows eSATA connectors on the back of a motherboard. eSATA uses shielded cable lengths up to 2 meters outside the PC and is hot pluggable. The beauty of eSATA is that it extends the SATA bus at full speed, so you're not limited to the meager 50 or 60 MBps of FireWire or USB.

Figure 9-27
eSATA connectors
(center; that's a
FireWire port
on the left)



SCSI: Still Around

Many specialized server machines and enthusiasts' systems use the SCSI technologies for various pieces of core hardware and peripherals, from hard drives to printers to high-end tape backup machines. SCSI is different from ATA in that SCSI devices connect together in a string of devices called a *chain*. Each device in the chain gets a SCSI ID to distinguish it from other devices on the chain. Last, the ends of a SCSI chain must be terminated. Let's dive into SCSI now, and see how SCSI chains, SCSI IDs, and termination all work.

SCSI is an old technology dating from the late 1970s, but it has been continually updated. SCSI is faster than ATA (though the gap is closing fast), and until SATA arrived SCSI was the only choice for anyone using RAID (see the "RAID" section a little later). SCSI is arguably fading away, but it still deserves some mention.

SCSI Chains

SCSI manifests itself through a SCSI chain, a series of SCSI devices working together through a host adapter. The host adapter provides the interface between the SCSI chain and the PC. Figure 9-28 shows a typical PCI host adapter. Many techs refer to the host adapter as the *SCSI controller*, so you should be comfortable with both terms.

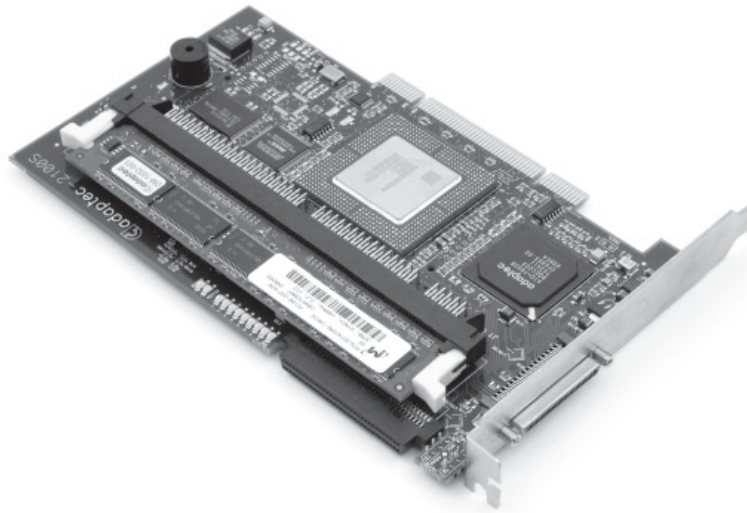


Figure 9-28 SCSI host adapter

All SCSI devices can be divided into two groups: internal and external. Internal SCSI devices are attached inside the PC and connect to the host adapter through the latter's internal connector. Figure 9-29 shows an internal SCSI device, in this case a CD-ROM drive. External devices hook to the external connector of the host adapter. Figure 9-30 is an example of an external SCSI device.



Figure 9-29 Internal SCSI CD-ROM



Figure 9-30 Back of external SCSI device

Internal SCSI devices connect to the host adapter with a 68-pin ribbon cable (Figure 9-31). This flat, flexible cable functions precisely like a PATA cable. Many external devices connect to the host adapter with a 50-pin high density (HD) connector. Figure 9-32 shows a host adapter external port. Higher end SCSI devices use a 68-pin high density (HD) connector.



Figure 9-31 Typical 68-pin ribbon cable

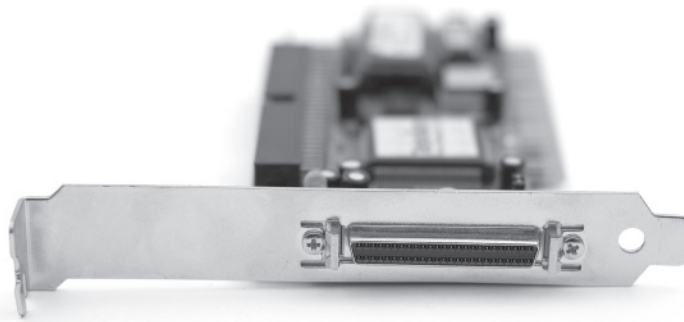
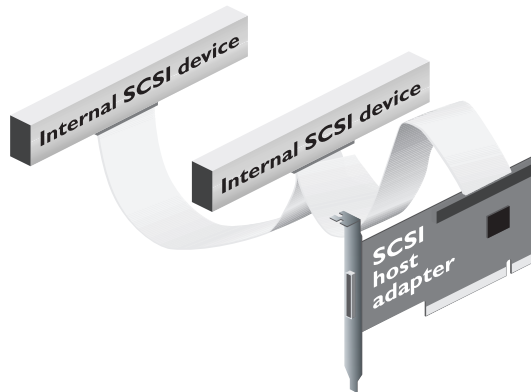


Figure 9-32 50-pin HD port on SCSI host adapter

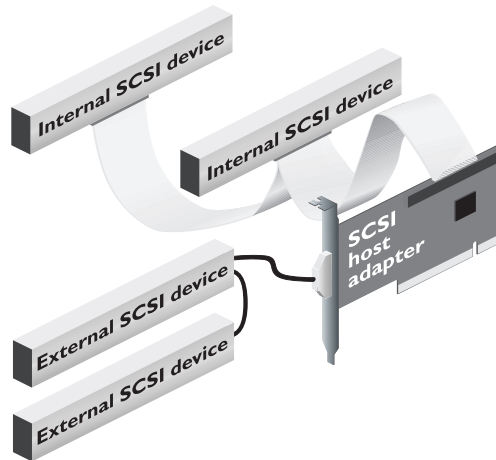
Multiple internal devices can be connected together simply by using a cable with enough connectors. Figure 9-33, for example, shows a cable that can take up to four SCSI devices, including the host adapter.

Figure 9-33
Internal SCSI chain
with two devices



Assuming the SCSI host adapter has a standard external port (some controllers don't have external connections at all), plugging in an external SCSI device is as simple as running a cable from device to controller. The external SCSI connectors are D-shaped, so you can't plug them in backward. As an added bonus, some external SCSI devices have two ports, one to connect to the host adapter and a second to connect to another SCSI device. The process of connecting a device directly to another device is called *daisy-chaining*. You can daisy-chain up to 15 devices to one host adapter. SCSI chains can be internal, external, or both (see Figure 9-34).

Figure 9-34
Internal and external
devices on one SCSI
chain



SCSI IDs

If you're going to connect a number of devices on the same SCSI chain, you must provide some way for the host adapter to tell one device from another. To differentiate devices, SCSI uses a unique identifier called the *SCSI ID*. The SCSI ID number can range from 0 to 15. SCSI IDs are similar to many other PC hardware settings in that a SCSI device can theoretically have any SCSI ID, as long as that ID is not already taken by another device connected to the same host adapter.



NOTE Old SCSI equipment allowed SCSI IDs from 0 to 7 only.

Some conventions should be followed when setting SCSI IDs. Typically, most people set the host adapter to 7 or 15, but you can change this setting. Note that there is no order for the use of SCSI IDs. It does not matter which device gets which number, and you can skip numbers. Restrictions on IDs apply only within a single chain. Two devices can have the same ID, in other words, as long as they are on different chains (Figure 9-35).

Figure 9-35

IDs don't conflict between separate SCSI chains.



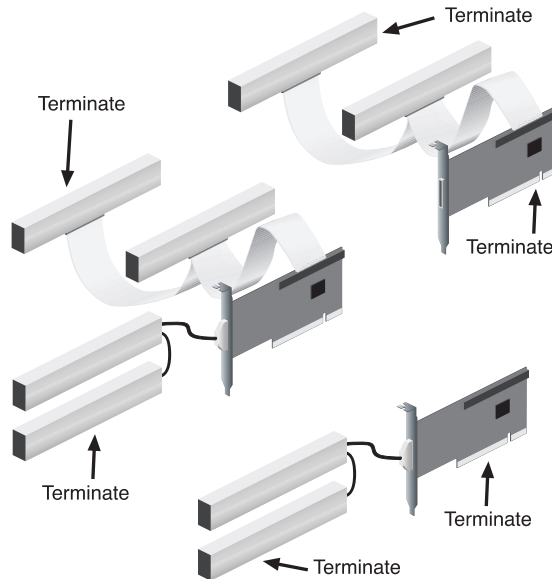
Every SCSI device has some method of setting its SCSI ID. The trick is to figure out how as you're holding the device in your hand. A SCSI device may use jumpers, dip switches, or even tiny dials; every new SCSI device is a new adventure as you try to determine how to set its SCSI ID.

Termination

Whenever you send a signal down a wire, some of that signal reflects back up the wire, creating an echo and causing electronic chaos. SCSI chains use *termination* to prevent this problem. Termination simply means putting something on the ends of the wire to prevent this echo. Terminators are usually pull-down resistors and can manifest themselves in many different ways. Most of the devices within a PC have the appropriate termination built in. On other devices, including SCSI chains and some network cables, you have to set termination during installation.

The rule with SCSI is that you *must* terminate *only* the ends of the SCSI chain. You have to terminate the ends of the cable, which usually means that you need to terminate the two devices at the ends of the cable. Do *not* terminate devices that are not on the ends of the cable. Figure 9-36 shows some examples of where to terminate SCSI devices.

Figure 9-36
Location of the
terminated devices



Because any SCSI device might be on the end of a chain, most manufacturers build SCSI devices that can self-terminate. Some devices will detect that they are on the end of the SCSI chain and will automatically terminate themselves. Most devices, however, require that you set a jumper or switch to enable termination (Figure 9-37).

Figure 9-37
Setting termination



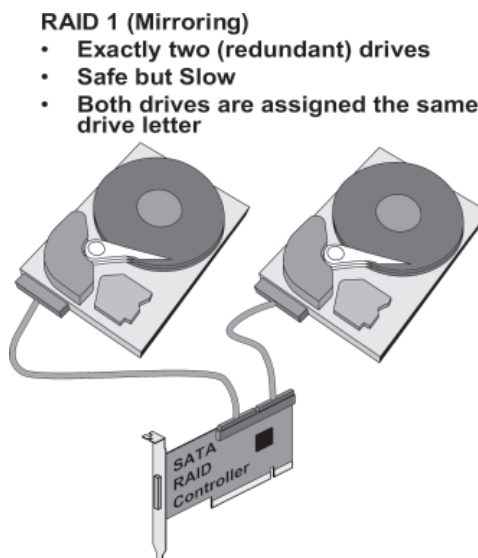
Protecting Data with RAID

Ask experienced techs, “What is the most expensive part of a PC?” and they’ll all answer in the same way: “It’s the data.” You can replace any single part of your PC for a few hundred dollars at most, but if you lose critical data—well, let’s just say I know of two small companies that went out of business just because they lost a hard drive full of data.

Data is king; data is your PC's *raison d'être*. Losing data is a bad thing, so you need some method to prevent data loss. Now, of course, you can do backups, but if a hard drive dies, you have to shut down the computer, reinstall a new hard drive, reinstall the operating system, and then restore the backup. There's nothing wrong with this as long as you can afford the time and cost of shutting down the system.

A better solution, though, would save your data if a hard drive died and enable you to continue working throughout the process. This is possible if you stop relying on a single hard drive and instead use two or more drives to store your data. Sounds good, but how do you do this? Well, first of all, you could install some fancy hard drive controller that reads and writes data to two hard drives simultaneously (Figure 9-38). The data on each drive would always be identical. One drive would be the primary drive and the other drive, called the *mirror* drive, would not be used unless the primary drive failed. This process of reading and writing data at the same time to two drives is called *disk mirroring*.

Figure 9-38
Mirrored drives

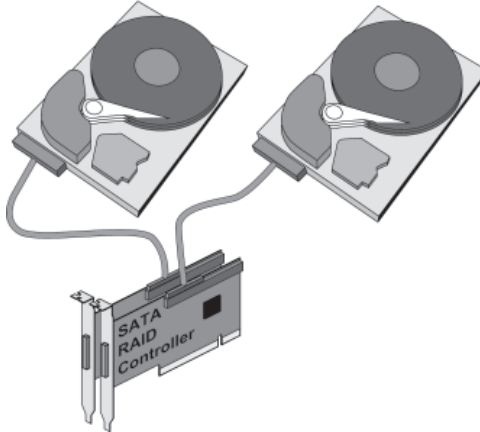


If you really want to make data safe, you can use two separate controllers for each drive. With two drives, each on a separate controller, the system will continue to operate, even if the primary drive's controller stops working. This super-drive mirroring technique is called *disk duplexing* (Figure 9-39). Disk duplexing is also much faster than disk mirroring because one controller does not write each piece of data twice.

Figure 9-39
Duplexing drives

RAID 1 (Duplexing)

- Exactly two (redundant) drives
- Faster and safer!
- Both drives are assigned the same drive letter

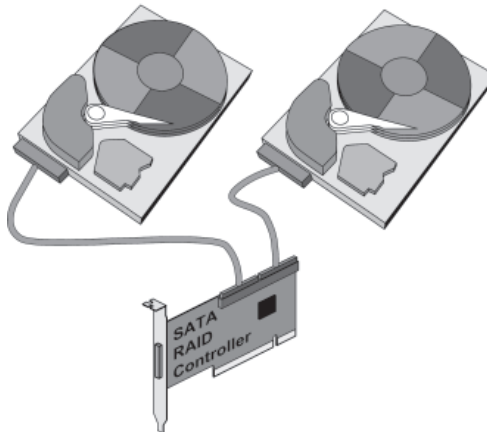


Even though duplexing is faster than mirroring, they both are slower than the classic one drive, one controller setup. You can use multiple drives to increase your hard drive access speed. *Disk striping* (without parity) means spreading the data among multiple (at least two) drives. Disk striping by itself provides no redundancy. If you save a small Microsoft Word file, for example, the file is split into multiple pieces; half of the pieces go on one drive and half on the other (Figure 9-40).

Figure 9-40
Disk striping

RAID 0 (Striping)

- Two or more (non-redundant) drives
- Fast, but not really safe
- Both drives are assigned the same drive letter



The one and only advantage of disk striping is speed—it is a fast way to read and write to hard drives. But if either drive fails, *all* data is lost. Disk striping is not something you should do—unless you’re willing to increase the risk of losing data to increase the speed at which your hard drives save and restore data.

Disk striping with parity, in contrast, protects data by adding extra information, called *parity data*, that can be used to rebuild data should one of the drives fail. Disk striping with parity requires at least three drives, but it is common to use more than three. Disk striping with parity combines the best of disk mirroring and plain disk striping. It protects data and is quite fast. The majority of network servers use a type of disk striping with parity.

RAID

A couple of sharp guys in Berkeley back in the 1980s organized the many techniques for using multiple drives for data protection and increasing speeds as the *random array of inexpensive disks* or *redundant array of independent disks (RAID)*. They outlined seven levels of RAID, numbered 0 through 6.



NOTE An array in the context of RAID refers to a collection of two or more hard drives.

- **RAID 0—Disk Striping** Disk striping requires at least two drives. It does not provide redundancy to data. If any one drive fails, all data is lost.
- **RAID 1—Disk Mirroring/Duplexing** RAID 1 arrays require at least two hard drives, although they also work with any even number of drives. RAID 1 is the ultimate in safety, but you lose storage space since the data is duplicated—you need two 100-GB drives to store 100 GB of data.
- **RAID 2—Disk Striping with Multiple Parity Drives** RAID 2 was a weird RAID idea that never saw practical use. Unused, ignore it.
- **RAID 3 and 4—Disk Striping with Dedicated Parity** RAID 3 and 4 combined dedicated data drives with dedicated parity drives. The differences between the two are trivial. Unlike RAID 2, these versions did see some use in the real world but were quickly replaced by RAID 5.
- **RAID 5—Disk Striping with Distributed Parity** Instead of dedicated data and parity drives, RAID 5 distributes data and parity information evenly across all drives. This is the fastest way to provide data redundancy. RAID 5 is by far the most common RAID implementation and requires at least three drives. RAID 5 arrays effectively use one drive’s worth of space for parity. If, for example, you have three 200-GB drives, your total storage capacity is 400 GB. If you have four 200-GB drives, your total capacity is 600 GB.
- **RAID 6—Disk Striping with Extra Parity** If you lose a hard drive in a RAID 5 array, your data is at great risk until you replace the bad hard drive and rebuild the array. RAID 6 is RAID 5 with extra parity information. RAID 6 needs at least five drives, but in exchange you can lose up to two drives at the same time. RAID 6 is gaining in popularity for those willing to use larger arrays.

After these first RAID levels were defined, some manufacturers came up with ways to combine different RAIDs. For example, what if you took two pairs of striped drives and mirrored the pairs? You would get what is called RAID 0+1. Or what (read this carefully now) if you took two pairs of mirrored drives and striped the pairs? You then get what we call RAID 1+0 or what is often called RAID 10. Combinations of different types of single RAID are called Multiple RAID solutions. Multiple RAID solutions, while enjoying some support in the real world, are quite rare when compared to single RAID solutions RAID 0, 1, and 5.



NOTE There is actually a term for a storage system composed of multiple independent disks, rather than disks organized using RAID: *JBOD*, which stands for Just a Bunch of Disks (or Drives).

Implementing RAID

RAID levels describe different methods of providing data redundancy or enhancing the speed of data throughput to and from groups of hard drives. They do not say *how* to implement these methods. Literally thousands of different methods can be used to set up RAID. The method used depends largely on the desired level of RAID, the operating system used, and the thickness of your wallet.

The obvious starting place for RAID is to connect at least two hard drives in some fashion to create a RAID array. For many years, if you wanted to do RAID beyond RAID 0 and RAID 1, the only technology you could use was good-old SCSI. SCSI's chaining of multiple devices to a single controller made it a natural for RAID. SCSI drives make superb RAID arrays, but the high cost of SCSI drives and RAID-capable host adapters kept RAID away from all but the most critical systems—usually big file servers.

In the last few years, substantial leaps in ATA technology have made ATA a viable alternative to SCSI drive technology for RAID arrays. Specialized ATA RAID controller cards support ATA RAID arrays of up to 15 drives—plenty to support even the most complex RAID needs. In addition, the inherent hot-swap capabilities of serial ATA have virtually guaranteed that serial ATA will quickly take over the lower end of the RAID business. Personally, I think the price and performance of serial ATA mean SCSI's days are numbered.

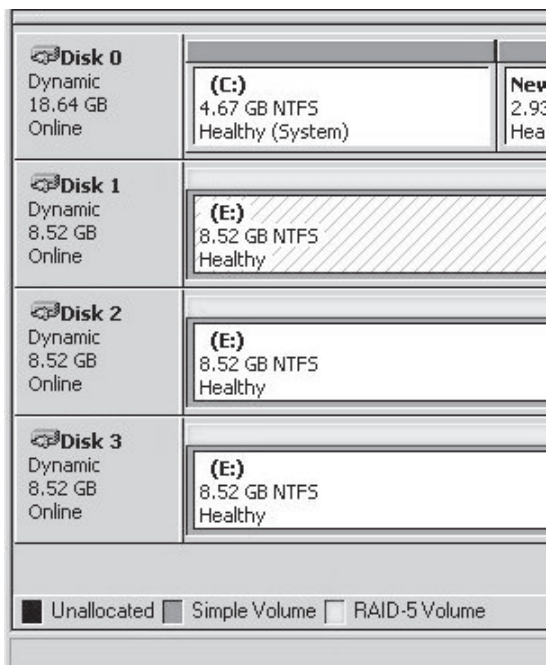
Once you have a number of hard drives, the next question is whether to use hardware or software to control the array. Let's look at both options.

Hardware versus Software

All RAID implementations break down into either hardware or software methods. Software is often used when price takes priority over performance. Hardware is used when you need speed along with data redundancy. Software RAID does not require special controllers—you can use the regular ATA controllers or SCSI host adapters to make a software RAID array. But you do need “smart” software. The most common software implementation of RAID is the built-in RAID software that comes with Windows 2000 Server and Windows Server 2003. The Disk Management program in these Windows Server versions can configure drives for RAID 0, 1, or 5, and it works with ATA or SCSI

(Figure 9-41). Disk Management in Windows 2000 Professional and Windows XP Professional, in contrast, can only do RAID 0.

Figure 9-41
Disk Management
in Windows
Server 2003



NOTE You can use Disk Management in Windows 2000 and Windows XP Professional to create RAID 1 and RAID 5 arrays, but you can use Disk Management only remotely on a server version of Windows (2000 Server or Server 2003). In other words, the capability is there, but Microsoft has limited the OS. If you want to use software RAID in Windows 2000 or XP (Home or Professional), you need to use a third-party tool to set it up.

Windows Disk Management is not the only software RAID game in town. A number of third-party software programs can be used with Windows or other operating systems.

Software RAID means the operating system is in charge of all RAID functions. It works for small RAID solutions but tends to overwork your operating system easily, creating slowdowns. When you *really* need to keep going, when you need RAID that doesn't even let the users know a problem has occurred, hardware RAID is the answer.

Hardware RAID centers around an *intelligent* controller—either a SCSI host adapter or an ATA controller that handles all of the RAID functions (Figure 9-42). Unlike a regular ATA controller or SCSI host adapter, these controllers have chips that know how to “talk RAID.”

Most RAID setups in the real world are hardware-based. Almost all of the many hardware RAID solutions provide *hot-swapping*—the ability to replace a bad drive without disturbing the operating system. Hot swapping is common in hardware RAID.

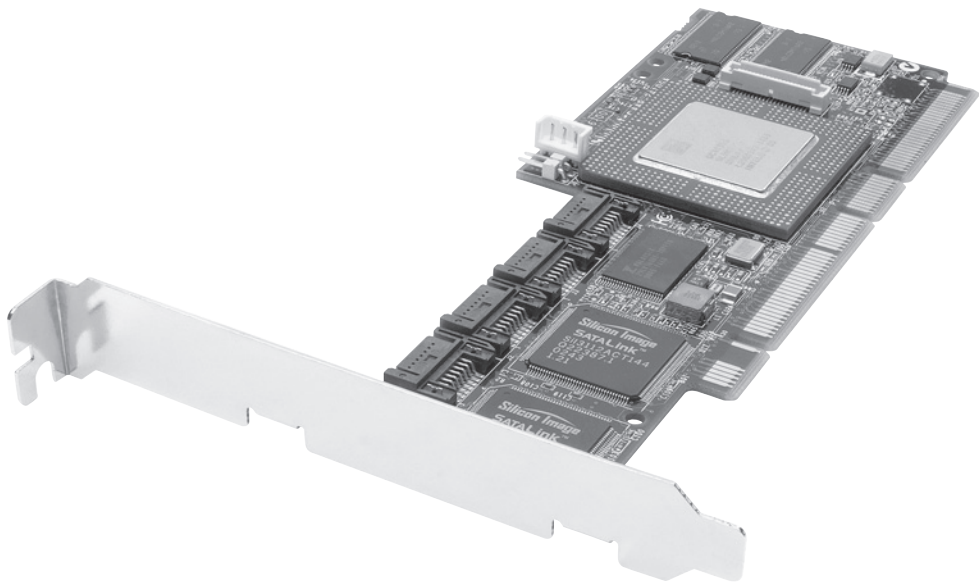


Figure 9-42 Serial ATA RAID controller

Hardware-based RAID is invisible to the operating system and is configured in several ways, depending on the specific chips involved. Most RAID systems have a special configuration utility in Flash ROM that you access after CMOS but before the OS loads. Figure 9-43 shows a typical firmware program used to configure a hardware RAID solution.

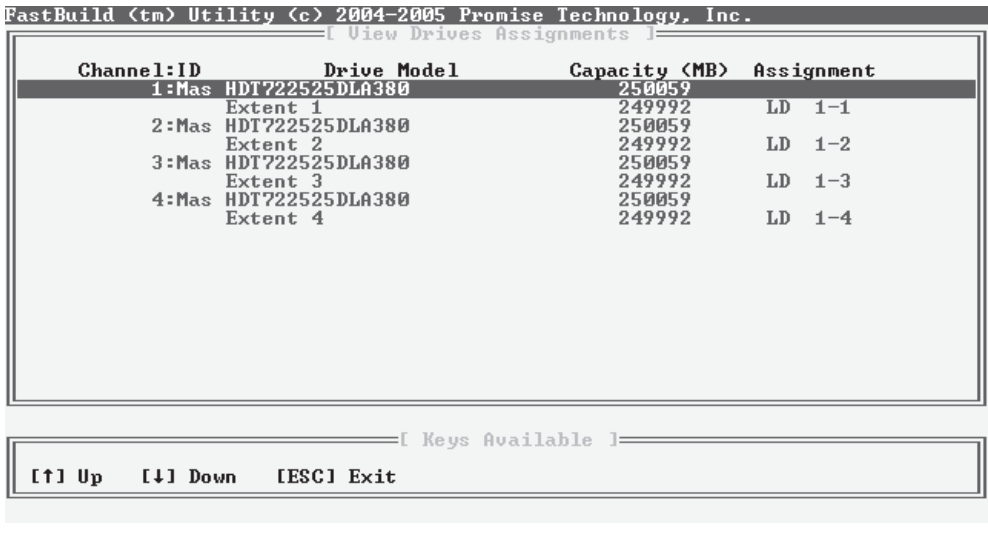


Figure 9-43 RAID configuration utility

Personal RAID

Due to drastic reductions in the cost of ATA RAID controller chips, in the last few years we've seen an explosion of ATA-based hardware RAID solutions built into mainstream motherboards. While this "ATA RAID on the motherboard" began with parallel ATA, the introduction of serial ATA made motherboards with built-in RAID extremely common.

These personal RAID motherboards might be quite common, but they're not used too terribly often given that these RAID solutions usually provide only RAID 0 or RAID 1. If you want to use RAID, spend a few extra dollars and buy a RAID 5-capable controller.



NOTE RAID controllers aren't just for internal drives; some models can handle multiple eSATA drives configured using any of the different RAID levels. If you're feeling lucky, you can create a RAID array using both internal and external SATA drives.

Managing Heat

Adding three or more fast hard drives into a cramped PC case can be a recipe for disaster to the unwary tech. All those disks spinning constantly create a phenomenal amount of heat. Heat kills PCs! You've got to manage the heat inside a RAID-enabled system or risk losing your data, drives, and basic system stability. The easiest way to do this is to add fans.

Open up your PC case and look for built-in places to mount fans. Note how many case fans you currently have installed and the sizes you can use. (Most cases use 80 mm fans, but 60 and 120 mm fans are common as well.) Jot down the particulars of your system and take a trip to the local PC store to check out the fans.

Before you get all fan-happy and grab the biggest and baddest fans to throw in your case, don't forget to think about the added noise level. Try to get a compromise between keeping your case cool enough and not causing early deafness!

The Future Is RAID

RAID has been with us for about 20 years, but until only recently it was the domain of big systems and deep pockets. During those 20 years, however, a number of factors have come together to make RAID a reality for both big servers and common desktop systems. Imagine a world where dirt-cheap RAID on every computer means no one ever again losing critical data. I get goose bumps just thinking about it!

Connecting Drives

Installing a drive is a fairly simple process if you take the time to make sure you've got the right drive for your system, configured the drive properly, and made a few quick tests to see if it's running properly. Since PATA, SATA, and SCSI have different cabling requirements, we'll look at each of these separately.

Choosing Your Drive

First, decide where you’re going to put the drive. Look for an open ATA connection. Is it PATA or SATA? Is it a dedicated RAID controller? Many motherboards with built-in RAID controllers have a CMOS setting that enables you to turn the RAID on or off (Figure 9-44). Do you have the right controller for a SCSI drive?

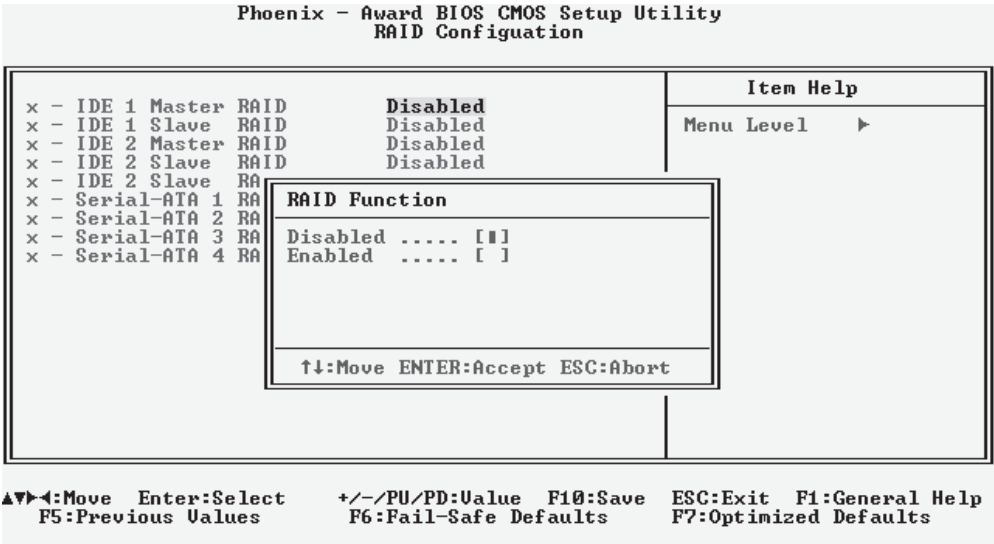


Figure 9-44 Settings for RAID in CMOS

Second, make sure you have room for the drive in the case. Where will you place it? Do you have a spare power connector? Will the data and power cables reach the drive? A quick test fit is always a good idea.

Don’t worry about PIO Modes and DMA—a new drive will support anything your controller wants to do.

Jumpers and Cabling on PATA Drives

If you have only one hard drive, set the drive’s jumpers to master or standalone. If you have two drives, set one to master and the other to slave. Figure 9-45 shows a close-up of a PATA hard drive showing the jumpers.

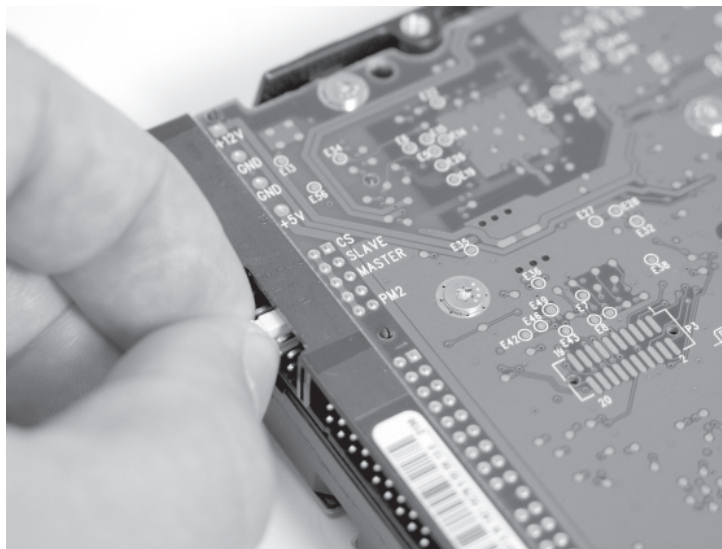
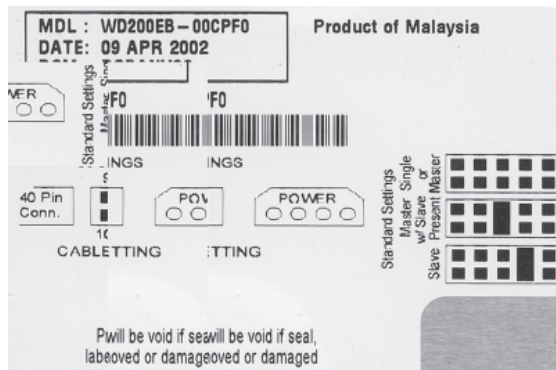


Figure 9-45 Master/slave jumpers on a hard drive

At first glance, you might notice that the jumpers aren't actually labeled *master* and *slave*. So how do you know how to set them properly? The easiest way is to read the front of the drive; most drives have a diagram on the housing that explains how to set the jumpers properly. Figure 9-46 shows the front of one of these drives, so you can see how to set the drive to master or slave.

Figure 9-46
Drive label showing
master/slave settings



Hard drives may have other jumpers that may or may not concern you during installation. One common set of jumpers is used for diagnostics at the manufacturing

plant or for special settings in other kinds of devices that use hard drives. Ignore them. They have no bearing in the PC world. Second, many drives provide a third setting, which is used if only one drive connects to a controller. Often, master and single drive are the same setting on the hard drive, although some hard drives require separate settings. Note that the name for the single drive setting varies among manufacturers. Some use Single; others use 1 Drive or Standalone.

Many current PATA hard drives use a jumper setting called *cable select*, rather than master or slave. As the name implies, the position on the cable determines which drive will be master or slave: master on the end, slave in the middle. For cable select to work properly with two drives, both drives must be set as cable select and the cable itself must be a special cable-select cable. If you see a ribbon cable with a pinhole through one wire, watch out! That's a cable-select cable.



NOTE Most of the high speed ATA/66/100/133 cables support cable select—try one and see!

If you don't see a label on the drive that tells you how to set the jumpers, you have several options. First, look for the drive maker's Web site. Every drive manufacturer lists its drive jumper settings on the Web, although it can take a while to find the information you want. Second, try phoning the hard drive maker directly. Unlike many other PC parts manufacturers, hard drive producers tend to stay in business for a long period of time and offer great technical support.

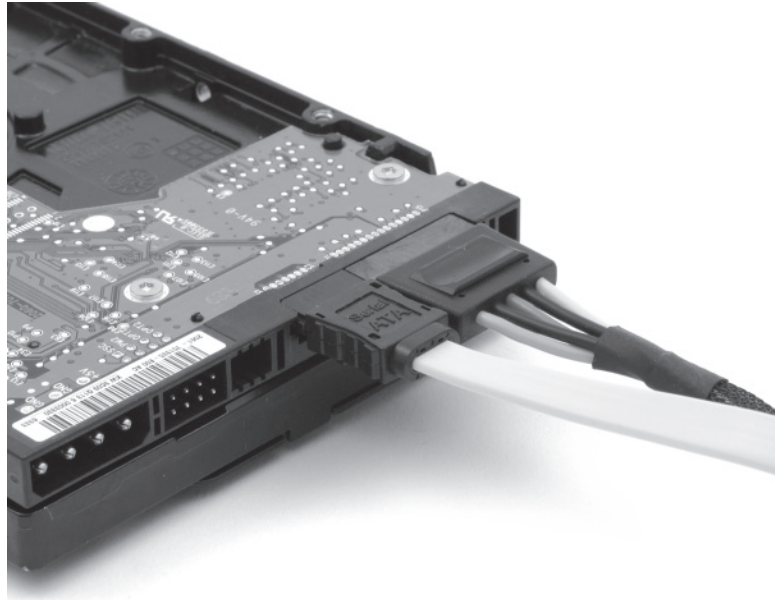
Hard drive cables have a colored stripe that corresponds to the number-one pin—called *Pin 1*—on the connector. You need to make certain that Pin 1 on the controller is on the same wire as Pin 1 on the hard drive. Failing to plug in the drive properly will also prevent the PC from recognizing the drive. If you incorrectly set the master/slave jumpers or cable to the hard drives, you won't break anything; it just won't work.

Finally, you need to plug a Molex connector from the power supply into the drive. All modern PATA drives use a Molex connector.

Cabling SATA Drives

Installing SATA hard drives is even easier than installing IDE devices due to the fact that there's no master, slave, or cable select configuration to mess with. In fact, there are no jumper settings to worry about at all, as SATA supports only a single device per controller channel. Simply connect the power and plug in the controller cable as shown in Figure 9-47—the OS automatically detects the drive and it's ready to go! The keying on SATA controller and power cables makes it impossible to install either incorrectly.

Figure 9-47
Properly connected
SATA cable



The biggest problem with SATA drives is that many motherboards come with four or more. Sure, the cabling is easy enough, but what do you do when it comes time to start the computer and the system is trying to find the right hard drive to boot up! That's where CMOS comes into play.

Connecting SCSI Drives

Connecting SCSI drives requires three things. You must use a controller that works with your drive. You need to set unique SCSI IDs on the controller and the drive. Finally, you need to connect the ribbon cable and power connections properly. With SCSI, you need to attach the data cable correctly. You can reverse a PATA cable, for example, and nothing happens except the drive doesn't work. If you reverse a SCSI cable, however, you can seriously damage the drive. Just as with PATA cables, Pin 1 on the SCSI data cable must go to Pin 1 on both the drive and the host adapter.

BIOS Support: Configuring CMOS and Installing Drivers

Every device in your PC needs BIOS support, and the hard drive controllers are no exception. Motherboards provide support for the ATA hard drive controllers via the system BIOS, but they require configuration in CMOS for the specific hard drives attached. SCSI drives require software drivers or firmware on the host adapter.

In the old days, you had to fire up CMOS and manually enter CHS information whenever you installed a new ATA drive to ensure the system saw the drive. Today, this process still takes place, but it's much more automated. Still, there's plenty to do in CMOS when you install a new hard drive.

CMOS settings for hard drives vary a lot among motherboards. The following information provides a generic look at the most common settings, but you'll need to look at your specific motherboard manual to understand all the options available.

Configuring Controllers

As a first step in configuring controllers, make certain they're enabled. It's easy to turn off controllers in CMOS, and many motherboards turn off secondary ATA controllers by default. Scan through your CMOS settings to locate the controller on/off options (see Figure 9-48 for typical settings). This is also the time to check whether your on-board RAID controllers work in both RAID and non-RAID settings.

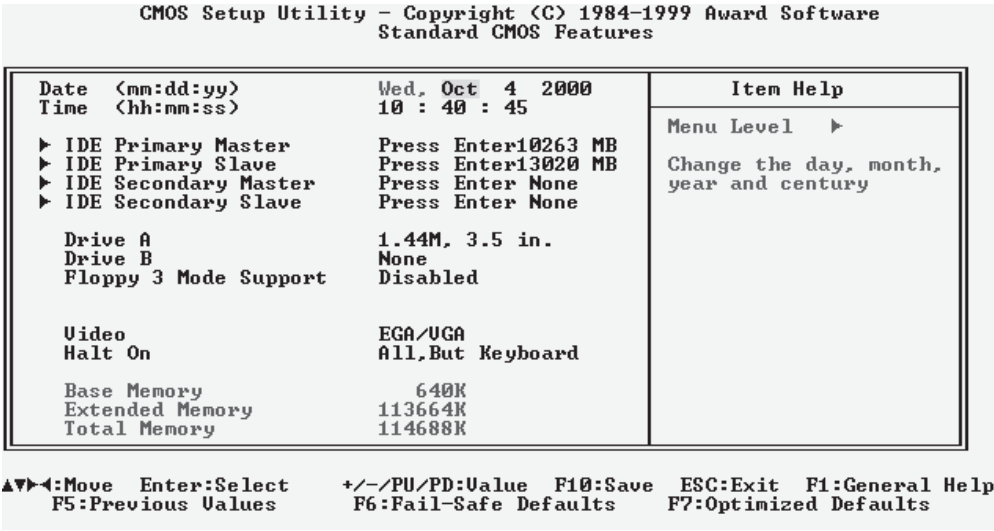


Figure 9-48 Typical controller settings in CMOS

Autodetection

If the controllers are enabled and the drive is properly connected, the drive should appear in CMOS through a process called *autodetection*. Autodetection is a powerful and handy feature, but it seems every CMOS has a different way to manifest it, and how it is manifested may affect how your computer decides which hard drive to try to boot when you start your PC.

One of your hard drives stores the operating system needed when you boot your computer, and your system needs a way to know where to look for this operating system. The traditional BIOS supported a maximum of only four ATA drives on two controllers, called the *primary controller* and the *secondary controller*. The BIOS looked for the master drive on the primary controller when the system booted up. If you used only one controller, you used the primary controller. The secondary controller was used for CD-ROMs, DVDs, or other non-bootable drives.

Older CMOS made this clear and easy, as shown in Figure 9-49. When you booted up, the CMOS would query the drives—a process called *autodetection*—and whatever drives the CMOS saw would show up here. Some even older CMOS had a special menu option called Autodetect that you had to run to see the drives in this screen. There are places for up to four devices—notice not all of them actually have a device.

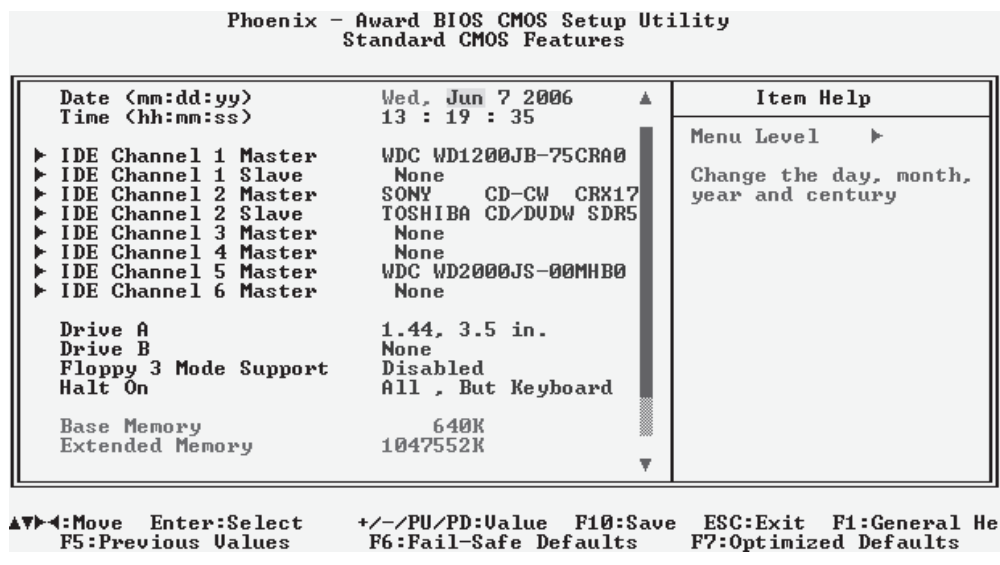


Figure 9-49 Old standard CMOS settings

The autodetection screen indicated that you installed a PATA drive correctly. If you installed a hard drive on the primary controller as master, but messed up the jumper and set it to slave, it would show up in the autodetection screen as the slave. If you had two drives and set them both to master, one drive or the other (or sometimes both) didn't appear, telling you that something was messed up in the physical installation. If you forgot to plug in the ribbon cable or the power, the drives wouldn't autodetect.

SATA messed up the autodetection happiness. There's no such thing as master, slave, or even primary and secondary controller in the SATA world. To get around this, motherboards with PATA and SATA today use a numbering system—and every motherboard uses its own numbering system! One common numbering method uses the term *channels* for each controller. The first boot device is channel 1, the second is channel 2, and so on. PATA channels may have a master and a slave, but a SATA channel has only a master, as SATA controllers support only one drive. So instead of names of drives, you see numbers. Take a look at Figure 9-50.

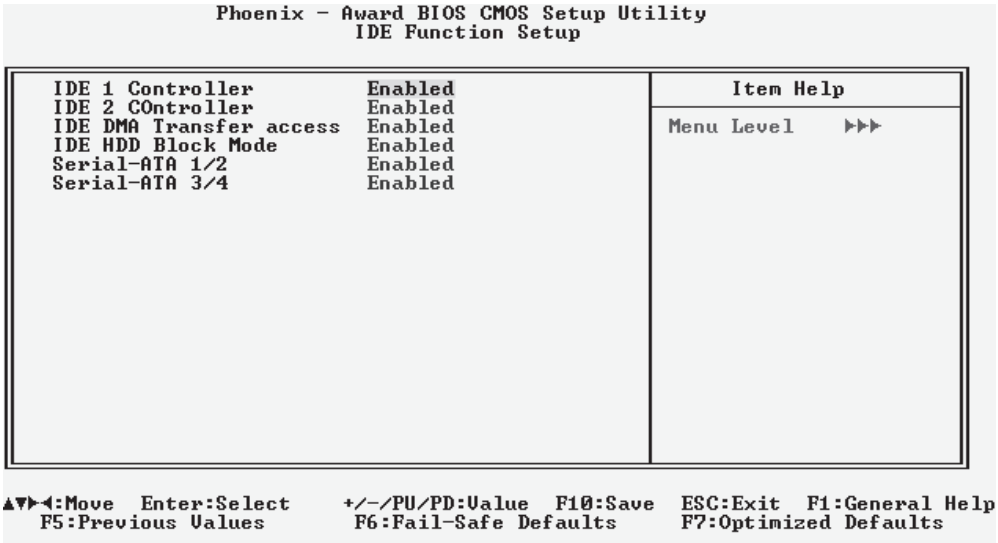


Figure 9-50 New standard CMOS features

Whew! Lots of hard drives! This motherboard supports the traditional four PATA drives, but it also supports four SATA drives. Each controller is assigned a number—note that channel 1 and channel 2 have master/slave settings, and that’s how you know channel 1 and 2 are the PATA drives. Channels 3 through 6 are SATA, even though the listing says *master*. (SATA’s still somewhat new, and a CMOS using incorrect terms like *master* is common.)

Boot Order

If you want your computer to run, it’s going to need an operating system to boot. While the PCs of our forefathers (those of the 1980s and early 1990s) absolutely required you to put the operating system on the primary master, most BIOS makers by 1995 enabled you to put the OS on any of the four drives and then tell the system through CMOS which hard drive to boot. Additionally, you may need to boot from a floppy, CD-ROM, or even a thumb drive at times. CMOS takes care of this by enabling you to set a *boot order*.

Figure 9-51 shows a typical boot order screen. It has a first, second, and third boot option. Many users like to boot first from floppy or CD-ROM, and then from a hard drive. This enables them to put in a bootable floppy or CD-ROM if they’re having problems with the system. Of course, you can set it to boot first from your hard drive and then go into CMOS and change it when you need to—it’s your choice.

Figure 9-51
Boot order

▶ Hard Disk Boot Priority	Press Enter
First Boot Device	Floppy
Second Boot Device	Hard Disk
Third Boot Device	CDROM
Boot Other Device	Enabled

Most modern CMOS lump the hard drive boot order onto a second screen. This screen works like an autodetect in that it shows only actual hard drives attached. This beats the heck out of guessing!

Device Drivers

Devices that do not get BIOS via the system BIOS routines naturally require some other source for BIOS. For ATAPI devices and many SATA controllers, the source of choice is software device drivers, but both technologies have a couple of quirks you should know about.

ATAPI Devices and BIOS

ATAPI drives plug into an ATA controller on the motherboard and follow the same conventions on cabling and jumpers used by PATA hard drives. In fact, all current CMOS setup utilities *seem* to autodetect CD-media ATAPI drives. If you go into CMOS after installing a CD-ROM drive as master on the secondary IDE controller, for example, the drive will show up just fine, as in Figure 9-52.

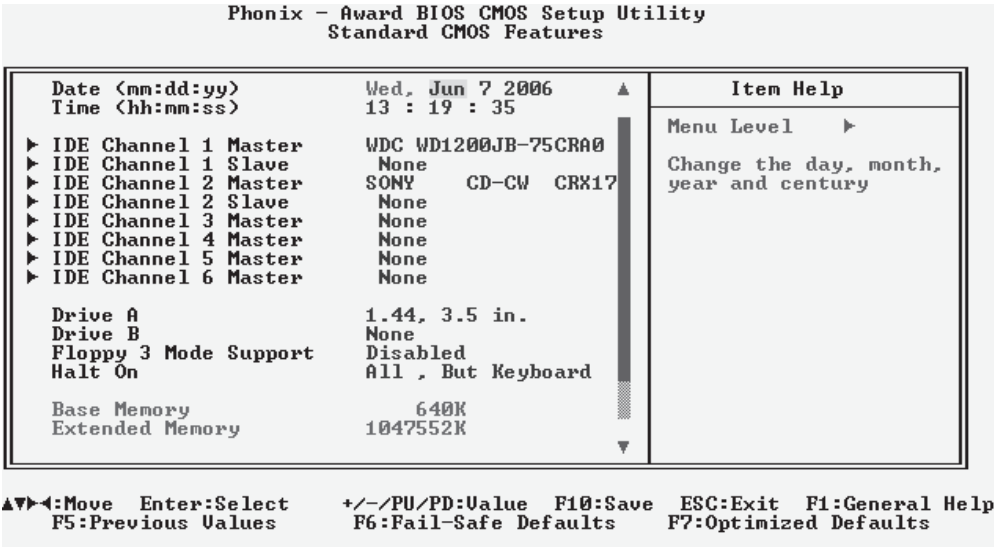


Figure 9-52 CMOS screen showing a CD-ROM drive detected

The reporting of installed CD-media drives in CMOS serves two purposes. First, it tells the technician that he or she has good connectivity on the ATAPI drive. Second, it shows that you have the option to boot to CD media, such as a Windows XP disc. What it doesn't do, however, is provide true BIOS support for that drive! That has to come with a driver loaded at boot-up.

Troubleshooting Hard Drive Installation

The best friend a tech has when it comes to troubleshooting hard drive installation is the autodetection feature of the CMOS setup utility. When a drive doesn't work, the biggest

question, especially during installation, is, “Did I plug it in correctly?” With autodetection, the answer is simple; if it doesn’t see the drives, something is wrong with the hardware configuration. Either a device has physically failed or, more likely, you didn’t give the hard drive power, plugged a cable in backwards, or messed up some other connectivity issue.

It takes three things to get a drive installed and recognized by the system: jumpers (PATA only), data cable, power, and CMOS setup recognizing the drive. If any of these steps are missed or messed up, you have a drive that simply doesn’t exist according to the PC! To troubleshoot hard drives, simply work your way through each step to figure out what went wrong.

First, set the drive to master, slave, standalone, or cable select, depending on where you decide to install the drive. If a drive is alone on the cable, set it to master or standalone. With two drives, one must be master and the other slave. Alternatively, you can set both drives to cable select and use a cable-select cable.

Second, the data cable must be connected to both the drive and controller, pin 1 to pin 1. Reversing the data cable at one end is remarkably easy to do, especially with the rounded cables. They obviously don’t have a big red stripe down the side to indicate the location of pin 1! If you can’t autodetect the drive, check the cabling.

Third, be sure to give the hard drive power. Most hard drives use a standard Molex connector. If you don’t hear the whirring of the drive, make certain you plugged in a Molex from the power supply, rather than from another source such as an otherwise disconnected fan. You’d be surprised how often I’ve seen that!

Fourth, you need to provide BIOS for the controller and the drive. This can get tricky as the typical CMOS setup program has a lot of hard drive options. Plus, you have an added level of confusion with RAID settings and non-integrated controllers that require software drivers.

Once you’ve checked the physical connections, run through these issues in CMOS. Is the controller enabled? Is the storage technology—LBA, INT13, ATA/ATAPI- 6—properly set up? Similarly, can the motherboard support the type of drive you’re installing? If not, you have a couple of options. You can flash the BIOS with an upgraded BIOS from the manufacturer or you can get a hard drive controller that goes into an expansion slot.

Finally, make certain with non-integrated hard drive controllers, such as those that come with many SATA drives, that you’ve installed the proper drivers for the controller. Driver issues can crop up with new, very large drives, and with changes in technology. Always check the manufacturer’s Web site for new drivers.

Beyond A+

Spindle (or Rotational) Speed

Hard drives run at a set spindle speed, measured in *revolutions per minute (RPM)*. Older drives run at the long-standard speed of 3,600 RPM, but new drives are hitting 15,000 RPM! The faster the spindle speed, the faster the controller can store and retrieve data. Here are the common speeds: 4,500, 5,400, 7,200, and 10,000 RPM.

Faster drives mean better system performance, but they can also cause the computer to overheat. This is especially true in tight cases, such as minitowers, and in

cases containing many drives. Two 4,500 RPM drives might run forever, snugly tucked together in your old case. But slap a hot new 10,000 RPM drive in that same case and watch your system start crashing right and left!

You can deal with these hotrod drives by adding drive bay fans between the drives or migrating to a more spacious case. Most enthusiasts end up doing both. Drive bay fans sit at the front of a bay and blow air across the drive. They range in price from \$10 to \$100 (U.S.) and can lower the temperature of your drives dramatically. Figure 9-53 shows a picture of a double-fan drive bay cooler.



Figure 9-53 Bay fans

Airflow in a case can make or break your system stability, especially when you add new drives that increase the ambient temperature. Hot systems get flaky and lock up at odd moments. Many things can impede the airflow—jumbled-up ribbon cables, drives squished together in a tiny case, fans clogged by dust or animal hair, and so on.

Technicians need to be aware of the dangers when adding a new hard drive to an older system. Get into the habit of tying off ribbon cables, adding front fans to cases when systems lock up intermittently, and making sure the fan(s) run well. Finally, if a client wants a new drive and his system is a tiny minitower with only the power supply fan to cool it off, be gentle, but definitely steer him to one of the slower drives!

Chapter Review Questions

1. John's system is running INT13 extensions. What is the maximum size hard drive it can support?
 - A. 504 GB
 - B. 137 GB

- C. 10 GB
 - D. 8.4 GB
2. How many IDE hard drives can you have on a system with two hard drive controllers?
- A. 1
 - B. 2
 - C. 3
 - D. 4
3. How do you differentiate two IDE drives on the same cable?
- A. The flat ribbon cable has a seven-wire twist that determines which is which.
 - B. Set jumpers on the individual drives to determine which is master and which is slave.
 - C. The IDE controller determines the hierarchy.
 - D. Both drives are considered equal.
4. What happens if you cable an IDE hard drive incorrectly?
- A. You can destroy that hard drive.
 - B. The data will be erased, but the hard drive will be okay.
 - C. The system will not be able to communicate with that hard drive.
 - D. Nothing. It doesn't matter how the cable is set up; it doesn't have the seven-wire twist.
5. John has just purchased an ATA/66-capable hard drive for his ATA/66-capable system. However, he notices that he forgot to get the special 80-wire cable. If he installs the ATA/66 drive with a 40-pin EIDE cable, what will happen?
- A. The drive will work, but will not run at the ATA/66 speed.
 - B. He might damage the motherboard.
 - C. He won't be able to install the drive, because the cables are incompatible.
 - D. He will not be able to run in ATA mode.
6. What is the maximum cable length of an internal serial ATA device ?
- A. 2 meters
 - B. 12 inches
 - C. 18 inches
 - D. 1 meter
7. Which of the following is part of the ATA-7 standard?
- A. Red controller connections on the motherboard
 - B. SATA
 - C. ATA/100
 - D. 1 meter maximum cable length

8. What is the maximum number of SATA drives you can have on a system?
 - A. One master, one slave
 - B. Two, with no master/slave distinction
 - C. Eight
 - D. There is no maximum other than the limitations of your motherboard.
9. Simon wants to connect his old PATA hard drive to the SATA controller in his new computer. What does he need to do this?
 - A. A SATA bridge to plug into the connector on the PATA drive
 - B. A PATA converter to plug into the SATA bridge on the controller
 - C. Nothing at all; he can just connect the PATA drive directly to the SATA controller
 - D. He can't do it; the SATA controller is not backward compatible with PATA drives
10. What do you call a series of SCSI devices working together through a host adapter?
 - A. A SCSI controller
 - B. A SCSI chain
 - C. RAID
 - D. Cabled SCSI

Answers

1. B. A system running INT13 extensions can support up to a 137-GB hard drive.
2. D. Each controller supports two drives.
3. B. Drives use master/slave jumpers to differentiate between the two drives.
4. C. Nothing will be damaged or lost—there just won't be any communication.
5. A. ATA/66 drives work fine with a 40-pin cable—they just won't run at ATA/66 speed.
6. D. The maximum cable length for an internal SATA device is 1 meter.
7. B. Serial ATA is part of the ATA-7 standard.
8. D. There is no maximum number of SATA drives you can have on a system, beyond the limits imposed by the number of ports on your motherboard/host card.
9. A. Simon needs a SATA bridge to plug into the connector on the PATA drive in order to connect his old PATA hard drive to the SATA controller.
10. B. A series of SCSI devices working together through a host adapter is a SCSI chain.