

# Mass Storage Technologies

In this chapter, you will learn how to

- Explain how hard drives work
  - Identify mass storage interface connections
  - Describe how to protect data with RAID
  - Describe hard drive installation
- 

Of all the hardware on a PC, none gets more attention—or gives more anguish—than mass storage drives. There’s a good reason for this: if a drive breaks, you lose data. As you probably know, when data goes, you have to redo work, restore from a backup, or worse, just kiss the data goodbye. It’s good to worry about data, because that data runs the office, maintains the payrolls, and stores the e-mail.

This chapter focuses on how drives work, beginning with the internal layout and organization of drives. You’ll look at the different types of drives used today and how they interface with the PC. 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. The chapter wraps up with an extensive discussion on how to install drives properly into a system. Let’s get started.

---



**NOTE** [Chapter 9](#), “Implementing Mass Storage,” 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 modern operating systems.

## Historical/Conceptual

### How Hard Drives Work

Hard drives come in two major types: the traditional type with moving parts; and a newer, more expensive technology with no moving parts. Let's look at both.

### Magnetic Hard Drives

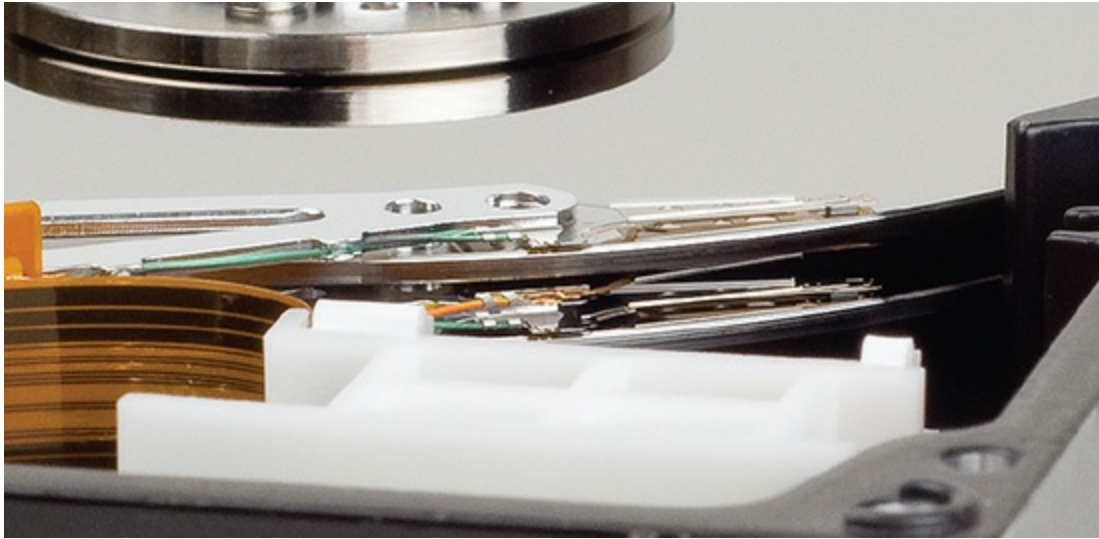
A traditional *hard disk drive (HDD)* is composed of individual disks, or *platters*, with read/write heads on actuator arms controlled by a servo motor—all contained in a sealed case that prevents contamination by outside air (see [Figure 8-1](#)).



---

**Figure 8-1** An enclosed HDD (top) and an opened HDD (bottom)

The aluminum platters are coated with a magnetic medium. Two tiny read/write heads service each platter, one to read the top of the platter and the other to read the bottom of the platter (see [Figure 8-2](#)). Each head has a bit-sized *transducer* to read or write to each spot on the drive. Many folks refer to traditional HDDs as *magnetic hard drives*, *rotational drives*, or sometimes *platter-based hard drives*.



**Figure 8-2** Read/write heads on actuator arms

## 1001

### Spindle (or Rotational) Speed

Hard drives run at a set *spindle speed*, with the spinning platters measured in *revolutions per minute (RPM)*. Older drives ran at a speed of 3600 RPM, and drives are sold with speeds up to 15,000 RPM. The faster the spindle speed, the faster the drive stores and retrieves data. By far the two most common speeds are 5400 and 7200 RPM. Higher performance drives (which are also far less common) run at 10,000 and 15,000 RPM.

Faster drives generally equate to better performance, but they also generate more noise and heat. Excess heat cuts the life of hard drives dramatically. A rise of 5 degrees (Celsius) may reduce the life expectancy of a hard drive by as much as two years. So even if replacing an old pair of 5400-RPM drives with a shiny new pair of 15,000-RPM drives doesn't generate enough heat to crash the entire system, it may severely put your storage investment at risk of a short life cycle.

You can deal with the warmth of these very fast 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 (USD) and

can lower the temperature of your drives dramatically. Some cases come with a bay fan built in (see [Figure 8-3](#)).



**Figure 8-3** Bay fan

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 (used by older storage systems, USB headers, and other attachments), 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 non-aerodynamic cables, adding front fans to cases when systems lock up intermittently, and making sure any fans run well. Finally, if a client wants a new drive for a system in a tiny minitower with only the power supply fan to cool it off, be gentle, but definitely steer the client to one of the slower drives!

## Form Factors

Magnetic hard drives are manufactured in two standardized form factors, 2.5-inch and 3.5-inch (see [Figure 8-4](#)). A desktop system can use either form factor size; most laptops use the 2.5-inch form factor.



---

**Figure 8-4** 2.5-inch drive stacked on top of a 3.5-inch drive

The form factor only defines size. The connections and the storage technology inside these drives can vary.

## Solid-State Drives

Booting up a computer takes time in part because a traditional hard drive needs to spin up before the read/write heads can retrieve data off the drive and load it into RAM. All of the moving metal parts of a platter-based hard drive use a lot of power, create a lot of heat, take up space, wear down over time, and take a lot of nanoseconds to get things done. A *solid-state drive* (SSD) addresses all of these issues nicely.

In technical terms, solid-state technology and devices are based on the combination of semiconductors and transistors used to create electrical components with no moving parts. That's a mouthful! In simple terms, SSDs use flash memory chips to store data instead of all those pesky metal spinning parts used in platter-based hard drives (see [Figure 8-5](#)).



**Figure 8-5** A solid-state drive

Solid-state technology is commonly used in desktop and laptop hard drives, memory cards, cameras, USB thumb drives, and other handheld

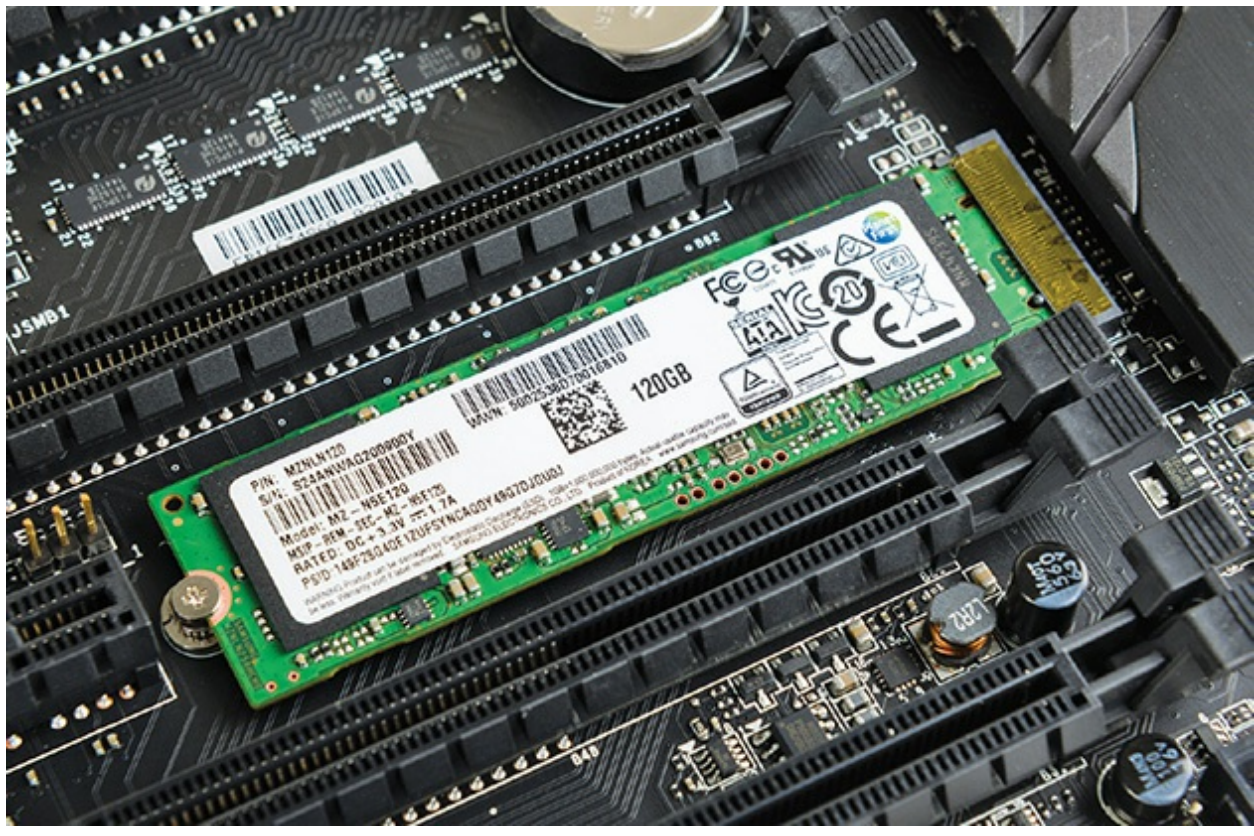


devices.

SSDs for personal computers come in one of three form factors: the 2.5-inch form factor previously mentioned and two flat form factors called *mSATA* and *M.2* (see [Figure 8-6](#)). mSATA and M.2 drives connect to specific mSATA or M.2 slots on motherboards (see [Figure 8-7](#)). Many current motherboards offer two or more M.2 slots.



**Figure 8-6** M.2 SSD



**Figure 8-7** M.2 SSD installed in motherboard





**EXAM TIP** Although you can still buy mSATA cards as we go to print, the technology is definitely on its way out for both laptop and desktop computers, replaced by M.2. The latter standard is half the physical size and offers substantially better performance. The M.2 form factor is incorrectly referred to as M2 (with no dot) in CompTIA A+ 1001 exam objective 3.4.

M.2 slots come in a variety, keyed for different sorts of mass storage uses. The keys have a letter associated. M.2 slots that use Key B, Key M, or Keys B+M support mass storage devices, for example, like in [Figure 8-7](#). Other slots like Key A and Key E are used in wireless networking devices. The specifics of the keys are beyond the current A+ exam, but M.2 looks like it's here to stay, so you need to be aware of the variations.

SSDs use nonvolatile flash memory such as *NAND* that retains data when power is turned off or disconnected. (See [Chapter 10](#), “Essential Peripherals,” for the scoop on flash memory technology.)

## Cost

SSDs cost more than HDDs. Less expensive SSDs typically implement less reliable *multi-level cell (MLC)* memory technology in place of the more efficient *single-level cell (SLC)* technology to cut costs. The most popular type of memory technology in SSDs is *3D NAND*, a form of MLC that stacks cells vertically, providing increased density and capacity.

Solid-state drives operate internally by writing data in a scattershot fashion to high-speed flash memory cells in accordance with the rules contained in the internal SSD controller. That process is hidden from the operating system by presenting an electronic façade to the OS that makes the SSD appear to be a traditional magnetic hard drive.

## Performance Variables

There are three big performance metrics to weigh when you buy an SSD: how fast it can read or write long sequences of data stored in the same part of the drive, how fast it can read or write small chunks of data scattered

randomly around the drive, and how quickly it responds to a single request. The value of each metric varies depending on what kind of work the drive will do. Before we dive into how you should weigh each metric, let's look at how the storage industry measures the sequential read/write performance, random read/write performance, and latency of individual SSDs.

**Sequential Read/Write Performance** A common measure of a storage device's top speed is its *throughput*, or the rates at which it can read and write long sequences of data. We usually express a device's *sequential read* and *sequential write* throughput in megabytes per second (MBps). Most drives read a little faster than they write.

For context, traditional hard drives generally have sequential read/write speeds that top out at 200 MBps; SATA SSDs can hit 600 MBps; and NVMe SSDs roll at 2500 MBps or faster. These numbers are useful if you know your drives will frequently read and write huge files, but very few real-world systems do.

**Random Read/Write Performance** Because real-world drives rarely get to read and write huge files all day, we also look at a drive's *random read*, *random write*, and *mixed random* performance. Basically, we measure how many times per second a device can read or write small, fixed-size chunks of data at random locations on the drive.

The labels for these measurements often reflect the size of the data chunk (usually 4 KB), so you may see them called *4K Read*, *4K Random Write*, *4K Mixed*, and so on. These measurements are all typically expressed as a number of *input/output operations per second (IOPS)*, but you may also see them expressed in MBps. For context, traditional hard drives typically clock in at fewer than 150 IOPS, whereas the latest NVMe SSDs boast *hundreds of thousands* of IOPS.

**Latency** It's also useful to look at a drive's *response time*, *access time*, or *latency*, which measures how quickly it responds to a single request. Latency is usually expressed in milliseconds (ms) or microseconds (μs). Low-latency storage is critical for high-performance file and database servers, but the latency of most modern drives is fine for general use. For context, traditional hard drives often have latencies under 20 ms, whereas SSDs commonly clock in well under 1 ms.

A lot of factors determine which combination of performance and price makes sense for a specific situation. A typical machine, for example, doesn't put a huge demand on the SSD. Users boot up the computer and then open an application or two and work. The quality of the SSD matters for boot-up time and application load, but the machine will rarely break a sweat after that. A workstation for high-end video editing, on the other hand, may read and write massive files for hours on end. A large file server may need to read and write thousands of tiny files a minute.

In practical terms, you can get by with a cheaper, lower-performing SSD in a general-use computer, but need to spend more for a higher performing SSD in demanding circumstances. When it comes to picking exactly which high-performance SSD, the throughput, IOPs, and latency metrics help you avoid overpaying for performance characteristics that don't matter for your use.

## Hybrid Hard Drives

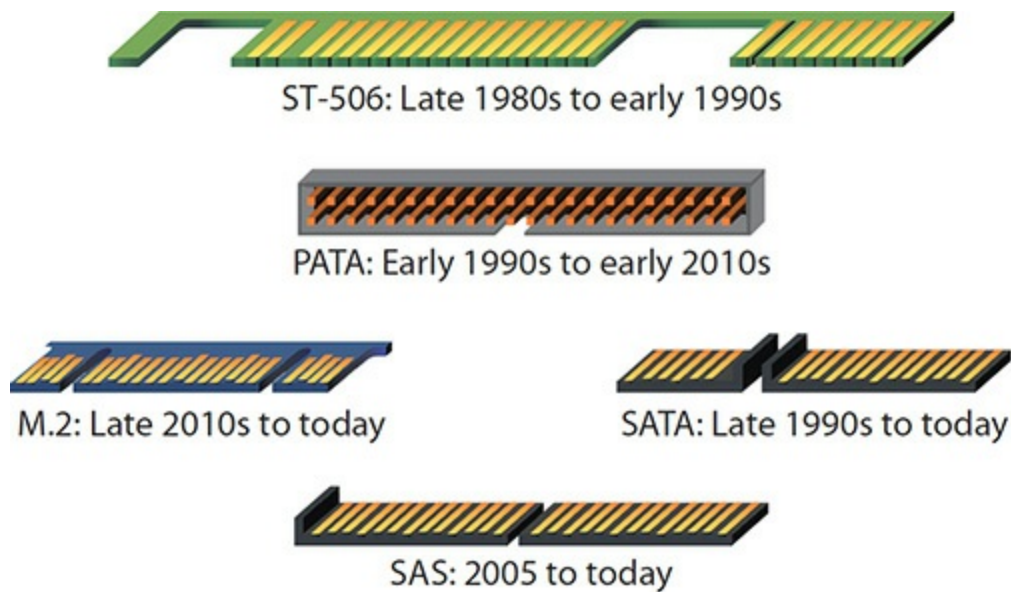
Windows supports *hybrid hard drives (HHDs)*, drives that combine flash memory and spinning platters to provide fast and reliable storage. (HHDs are also known as *SSHDs*.) The small SSD in these drives enables them to store the most accessed data in the flash memory to, for example, slash boot times and, because the platters don't have to spin as much, extend the battery life for portable computers.

Apple computers can use a *Fusion Drive*, which offers the same concept as a hybrid hard drive. The Fusion Drive separates the hard drive and SSD; macOS does all the work about deciding what should go in the SSD.

## Connecting Mass Storage

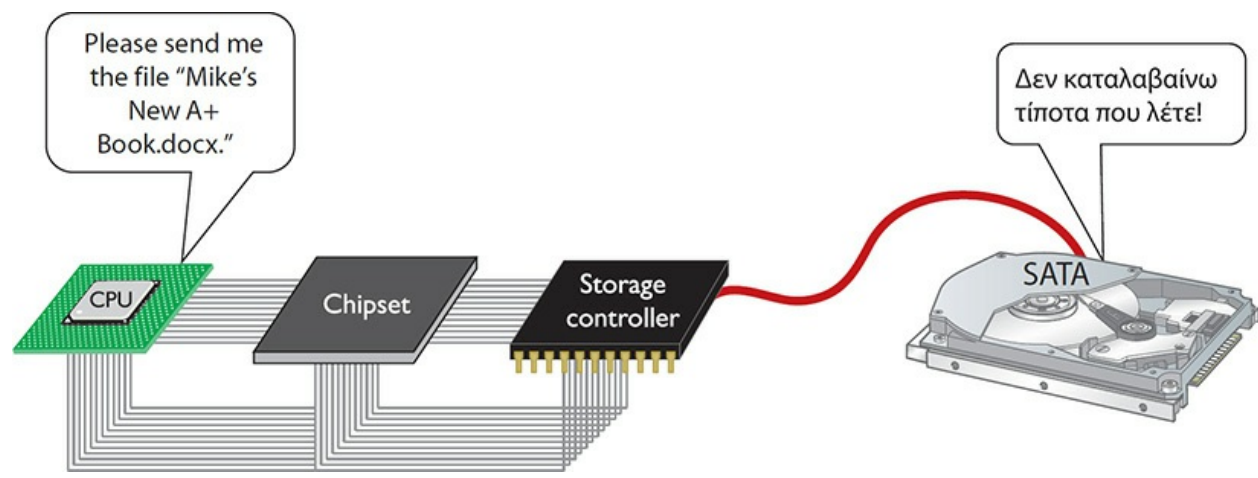
Setting up communication between a CPU and a mass storage drive requires two main items. First, there must be standardized physical connections between the CPU, the drive controller, and the physical drive. These connections must send data between these devices as quickly as possible while still retaining good security (see [Figure 8-8](#)).





**Figure 8-8** Standardized physical connections are essential.

Secondly, the CPU needs to use a standardized protocol, sort of like a special language, so it knows how to speak to the mass storage device to read and write data to the device (see [Figure 8-9](#)).



**Figure 8-9** We need a common language!

In most cases, the standards bodies that define both the physical connections as well as the language used for communications are the same organization. For the last 25+ years, the Storage Networking Industry Association's Small Form Factor (SFF) committee has defined mass storage standards, the most important to CompTIA A+ techs being ATA/ATAPI.



**NOTE** Check out [www.snia.org](http://www.snia.org) for a good source for mass storage standards.

The *advanced technology attachment (ATA)* standards started with version 1 way back in 1990, going through ATA/ATAPI version 7. Let's make it even easier, because only two versions of this standard have interest to techs: PATA and SATA. *Parallel ATA (PATA)* was introduced with ATA/ATAPI version 1. *Serial ATA (SATA)* was introduced with ATA/ATAPI version 7. Let's look at both standards.



**NOTE** ATA hard drives are often referred to as *integrated drive electronics (IDE)* drives. The term IDE 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 PATA drives. Many techs today use IDE only to refer to the older PATA standard.

## PATA

PATA drives are easily recognized by their data and power connections. PATA drives used unique 40-pin ribbon cables. These ribbon cables usually plugged directly into a system's motherboard. Note that the exam will call these *IDE cables*. [Figure 8-10](#) provides an example of a typical connection. All PATA drives used a standard Molex power connector (see [Figure 8-11](#)).



---

**Figure 8-10** PATA cable plugged into a motherboard



---

**Figure 8-11** Molex connector





**NOTE** The last ATA/ATAPI standard that addressed PATA provided support for very large hard drives (144 petabytes [PB], more than 144 million gigabytes) at speeds up to 133 megabytes per second (MBps). A single PATA ribbon cable could connect up to two PATA drives—including hard drives, optical drives, and tape drives—to a single ATA controller. You set jumpers on the drives to make one master and the other slave. (See the discussion on installation in the “Installing Drives” section later in this chapter for the full scoop.)

As a technology standard, ATA went through seven major revisions, each adding power, speed, and/or capacity to storage system capabilities. I could add 15 pages discussing the changes, but they’re not particularly relevant for modern techs. There is one feature added back then that we still use today, though, called *S.M.A.R.T.*

ATA/ATAPI version 3 introduced *Self-Monitoring, Analysis, and Reporting Technology* (*S.M.A.R.T.*), an internal drive program that tracks errors and error conditions within the drive. This information is stored in nonvolatile memory on the drive and can be examined externally with *S.M.A.R.T.* reader software. There are generic *S.M.A.R.T.* reading programs, and every drive manufacturer has software to get at the vendor-specific information being tracked. Regular usage of *S.M.A.R.T.* software will help you create a baseline of hard drive functionality to predict potential drive failures.

## SATA

For all its longevity as the mass storage interface of choice for the PC, parallel ATA had problems. First, the flat ribbon cables impeded airflow and could be a pain to insert properly. Second, the cables had a limited length, only 18 inches. Third, you couldn’t hot-swap PATA drives. You had to shut the computer down completely before installing or replacing a drive. Finally, the technology had simply reached the limits of what it could do in terms of throughput. It was time to revamp both the connection and the language for

ATA/ATAPI drives.

Serial ATA addressed these issues. SATA creates a point-to-point connection between the SATA device—magnetic hard drives, solid-state drives, optical media drives—and the SATA controller, the *host bus adapter (HBA)*. At a glance, SATA devices look identical to PATA devices. Take a closer look at the cable and power connectors, however, and you'll see significant differences (see [Figure 8-12](#)).



---

**Figure 8-12** SATA hard disk power (left) and data (right) cables

Because SATA devices send data serially instead of in parallel, the SATA interface needs far fewer physical wires—only 7 connectors instead of the 40 typical of PATA—resulting in much thinner cabling. 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 a PATA cable—about 40 inches (1 meter) instead of 18 inches. This facilitates drive installation in larger cases.

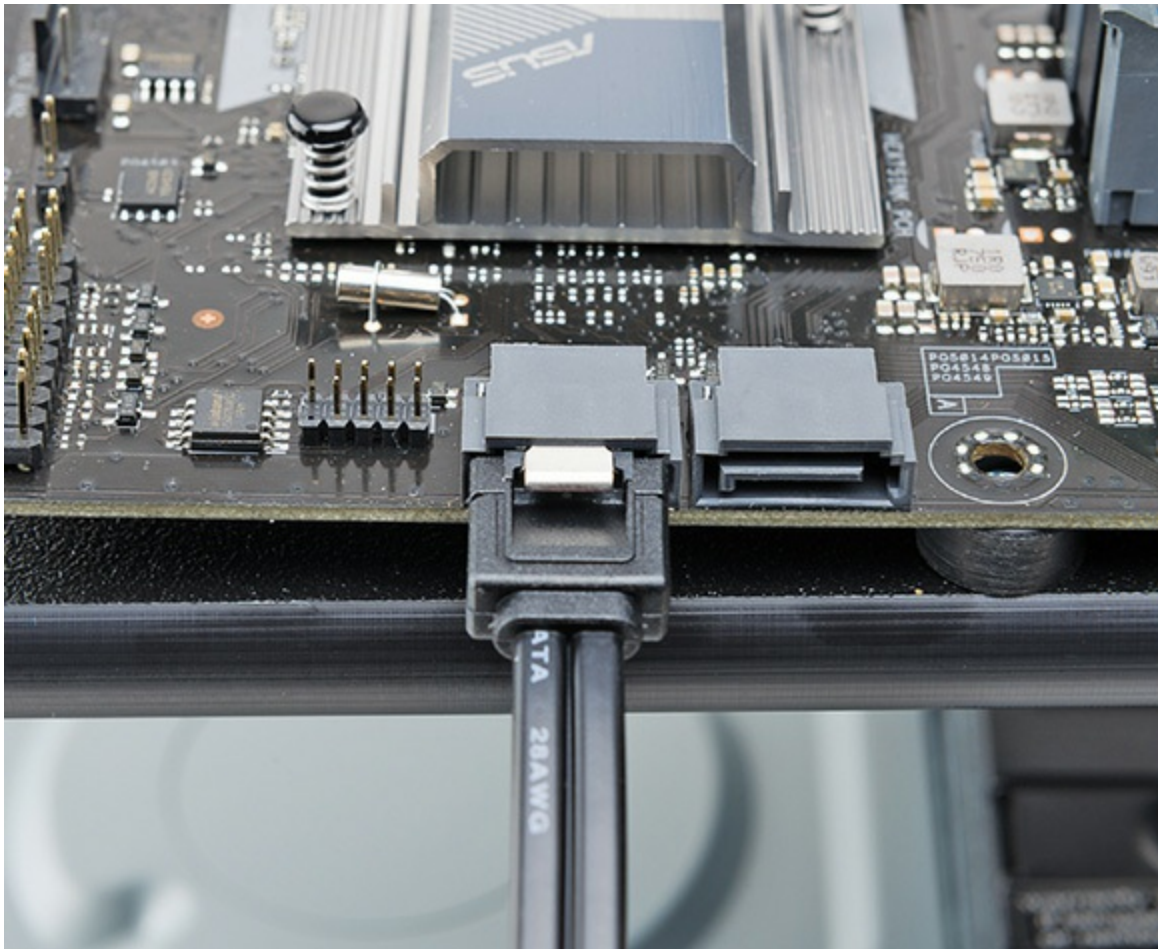


---

**EXAM TIP** The CompTIA A+ 1001 exam objectives refer to the 40-pin

PATA ribbon cable as an *IDE cable*. They're the same thing, so don't miss this one on the exam!

SATA did away with the two drives per cable of PATA. Each drive connects to one port. Further, there's no maximum number of drives—many motherboards today support up to eight SATA drives (see [Figure 8-13](#)). Want more? Snap in a SATA HBA and load 'em up!



**Figure 8-13** SATA cable plugged into typical motherboard (note the other available socket)

The biggest news about SATA 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, a SATA device's single stream of data moves much faster than the multiple streams of data coming from a parallel ATA



device—theoretically, up to 30 times faster. SATA drives come in three common SATA-specific varieties: *1.5 Gbps*, *3 Gbps*, and *6 Gbps*, which have a maximum throughput of 150 MBps, 300 MBps, and 600 MBps, respectively. It should be noted that if a system has an (external) eSATA port (discussed next), it will operate at the same revision and speed as the internal SATA ports.

---



**NOTE** Number-savvy readers might have noticed a discrepancy between the names and throughput of SATA drives. After all, SATA 1.0’s 1.5-Gbps throughput translates to 192 MBps, a lot higher than the advertised speed of a “mere” 150 MBps. The encoding scheme used on SATA drives takes about 20 percent of the transferred bytes as overhead, leaving 80 percent for pure bandwidth.

SATA 2.0’s 3-Gbps drive created all kinds of problems because 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 hard drives labeled “SATA II” rather than 3 Gbps.

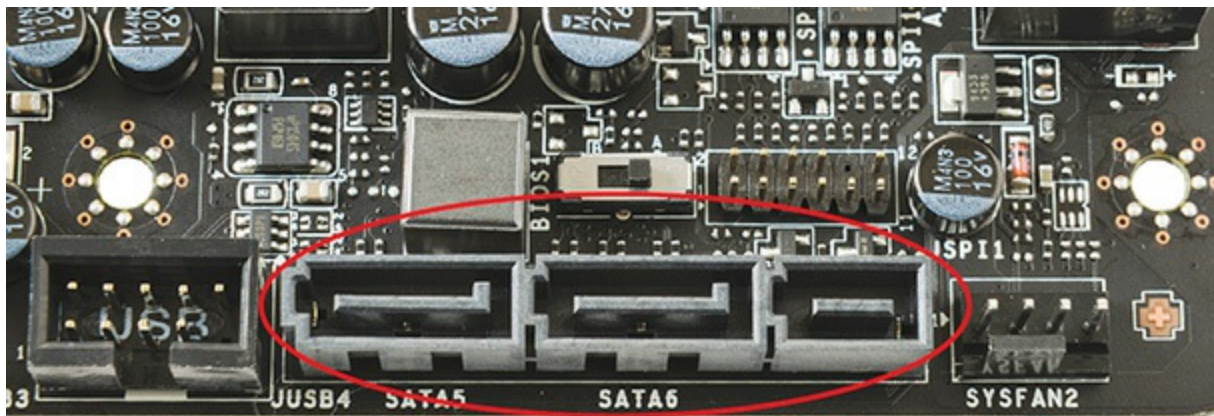
The SATA committee now goes by the name SATA-IO. In keeping with tradition, when SATA II speed doubled from 3 Gbps to 6 Gbps, two names were attached: SATA III and SATA 6 Gbps.

Connecting a mass storage device to a fully functioning and powered-up computer can result in disaster. The outcome may be as simple as the component not being recognized or as dire as a destroyed component or computer. Enter the era of the hot-swappable device. Hot-swapping entails two elements, the first being the capacity to plug a device into the computer without harming either. The second is that once the device is safely attached, it will be automatically recognized and become a fully functional component of the system. SATA handles hot-swapping just fine in modern systems (see “AHCI” later in the chapter for more details).

*SATA Express (SATAe)* or *SATA 3.2* ties capable drives directly into the PCI Express bus on motherboards. SATAe drops both the SATA link and transport layers, embracing the full performance of PCIe. The lack of

overhead greatly enhances the speed of SATA throughput, with each lane of PCIe 3.0 capable of handling up to 8 Gbps of data throughput. A drive grabbing two lanes, therefore, could move a whopping 16 Gbps through the bus. Without the overhead of earlier SATA versions, this translates as 2000 MBps!

SATAe has unique connectors (see [Figure 8-14](#)) but provides full backward compatibility with earlier versions of SATA. Note that the center and left portions of the port look just like regular SATA ports? They function that way too, so you can plug two regular SATA drives into a SATAe socket. Feel free to upgrade your motherboard! Oh yeah, did I forget to mention that? You'll need a motherboard with SATAe support to take advantage of these superfast versions of SATA drives.



**Figure 8-14** SATAe connector



**EXAM TIP** Each SATA variety is named for the revision to the SATA specification that introduced it, with the exception of SATAe:

- SATA 1.0: 1.5 Gbps/150 MBps
- SATA 2.0: 3 Gbps/300 MBps
- SATA 3.0: 6 Gbps/600 MBps
- SATA 3.2: up to 16 Gbps/2000 MBps, also known as SATAe

SATA's ease of use has made it the choice for desktop system storage.

Most hard drives sold today are SATA drives.

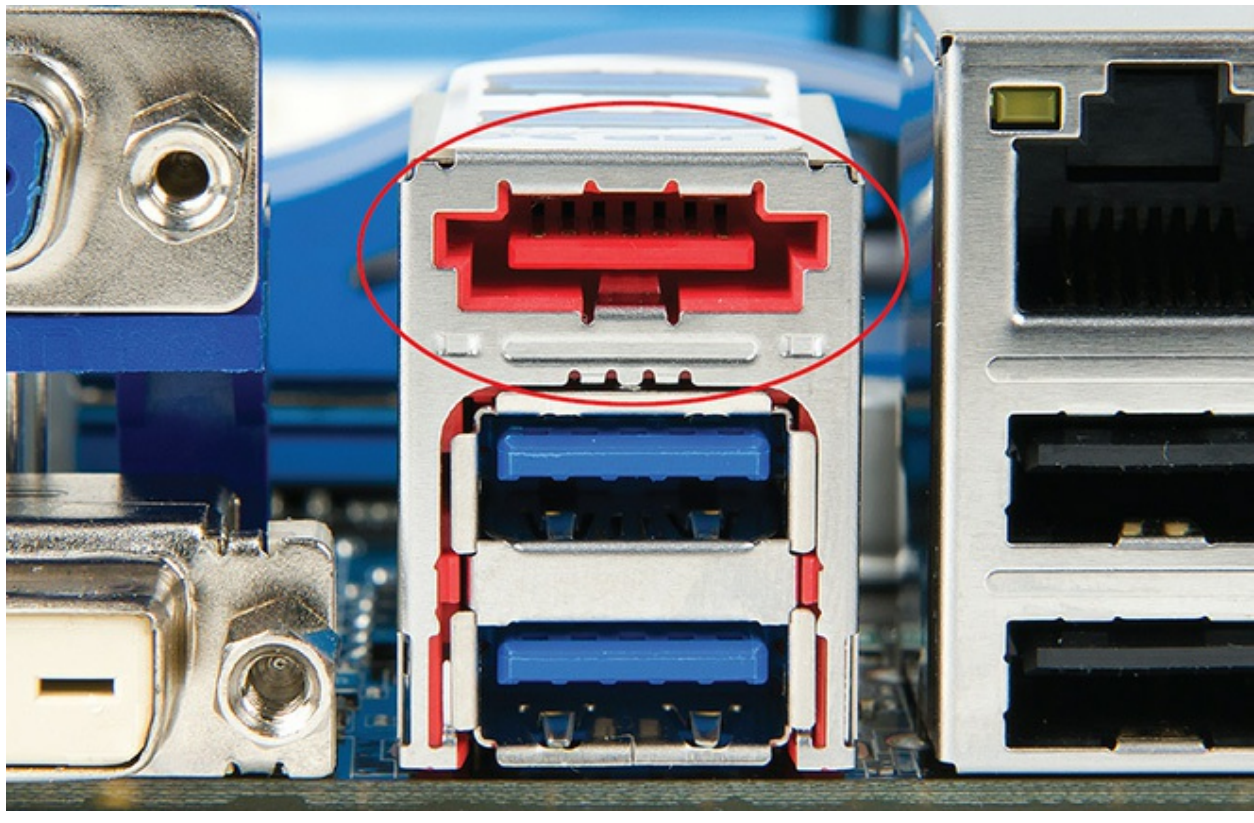
---



**NOTE** The SATA 3.3 (2016) revision increased supported drive sizes, among other things. The throughput speed of the interface did not increase.

## eSATA and Other External Drives

*External SATA (eSATA)* extended the SATA bus to external devices, as the name would imply. The eSATA drives used connectors that looked similar to internal SATA connectors, but were keyed differently so you couldn't mistake one for the other. [Figure 8-15](#) shows an eSATA connector on the back of a motherboard.



**Figure 8-15** eSATA connector



External SATA used shielded cable in lengths up to 2 meters outside the PC and was hot-swappable. eSATA extended the SATA bus at full speed, mildly faster than the fastest USB connection when it was introduced.

eSATA withered when USB 3.0 hit the market and quickly disappeared. You'll only find it today on very old systems and drive enclosures, and on the CompTIA A+ exam.



**EXAM TIP** The CompTIA A+ 1001 exam objectives mention *eSATA cards*, as in expansion cards you add to a system that doesn't have the connectors. You can certainly buy these today to support older, external mass storage enclosures.

Current *external enclosures* (the name used to describe the casing of external HDDs and SSDs) use the USB (3.0, 3.1, or C-type) ports or Thunderbolt ports for connecting external hard drives. [Chapter 10](#) goes into the differences among these types of ports in detail. The drives inside the enclosures are standard SATA HDDs or SSDs.



**EXAM TIP** Know your cable lengths:

- PATA: 18 inches
- SATA: 1 meter
- eSATA: 2 meters

## Refining Mass Storage Communication

The original ATA standard defined a very specific series of commands for the CPU to communicate with the drive controller. The current drive command sets are AHCI and NVMe. The SCSI command set is still around as well, though primarily in the server market.

# AHCI

Current versions of Windows support the *Advanced Host Controller Interface (AHCI)*, an efficient way to work with SATA HBAs. Using AHCI unlocks some of the advanced features of SATA, such as native command queuing and hot-swapping.

*Native command queuing (NCQ)* is a disk-optimization feature for SATA drives. It takes advantage of the SATA interface to achieve faster read and write speeds that are simply impossible with the old PATA drives. Also, while SATA supports hot-swapping ability, the motherboard and the operating system must also support this.

AHCI mode is enabled at the CMOS level (see “BIOS Support: Configuring CMOS and Installing Drivers” later in this chapter) and generally needs to be enabled before you install the operating system. Enabling it after installation will cause Windows to Blue Screen. How nice.

## **Successfully Switching SATA Modes Without Reinstalling**

You can attempt to switch to AHCI mode in Windows without reinstalling. This scenario might occur if a client has accidentally installed Windows in Legacy/IDE mode, for example, and finds that the new SSD he purchased requires AHCI mode to perform well.

First, back up everything before attempting the switch. Second, you need to run through some steps in Windows before you change the BIOS/UEFI settings. Windows 7 and 8/8.1 require manual changes to the Registry (the database that handles everything in Windows, covered in [Chapter 12](#), “Windows Under the Hood”). Windows 10 uses an elevated command prompt exercise with the `bcdedit` command. (The command line is covered in [Chapter 15](#), “Working with the Command-Line Interface.”)

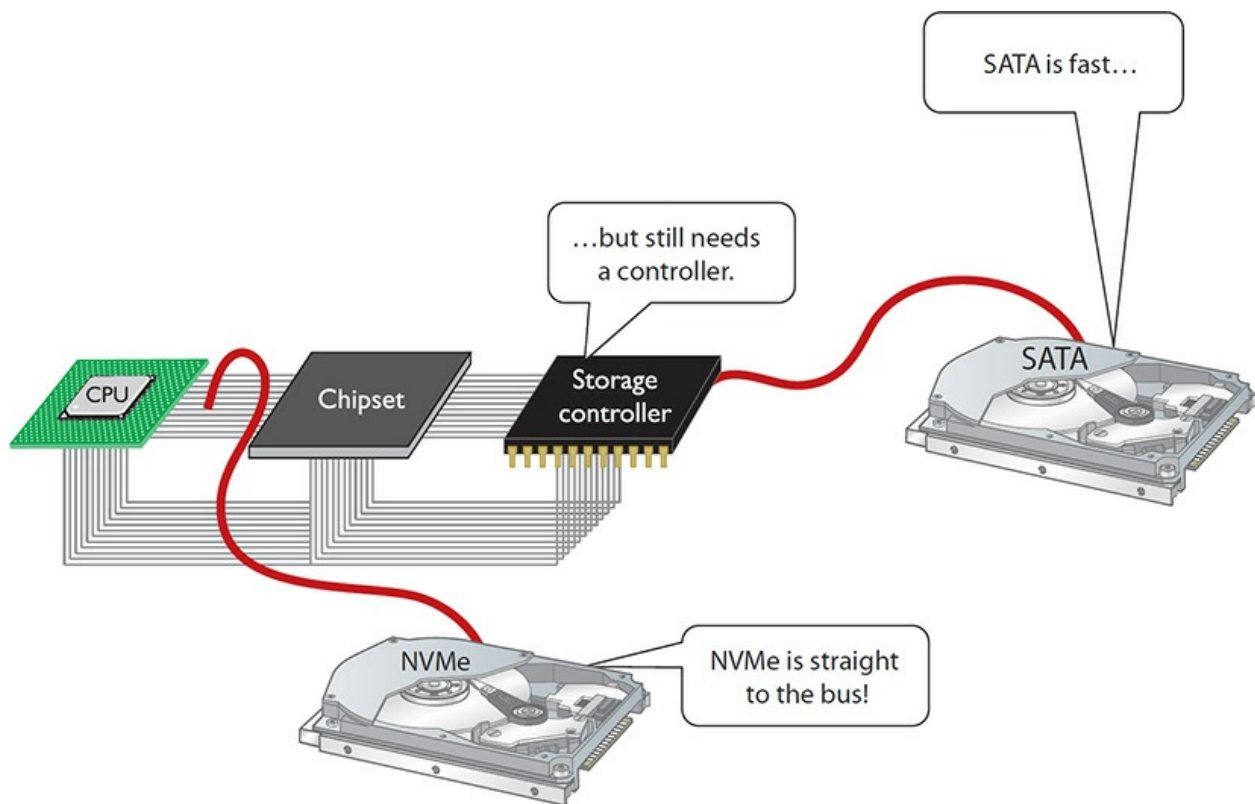
A quick Google search for “switch from ide to ahci windows” will reveal several excellent walkthroughs of the process for Windows 7/8/8.1 and Windows 10. Back everything up first!

When you plug a SATA drive into a running Windows computer that does not have AHCI enabled, the drive doesn’t appear automatically. With AHCI mode enabled, the drive should appear in Computer immediately, just what you’d expect from a hot-swappable device.

## NVMe

AHCI was designed for spinning SATA drives to optimize read performance as well as to effect hot-swappability. As a configuration setting, it works for many SSDs as well, but it's not optimal. That's because for an SSD to work with the operating system, the SSD has to include some circuitry that the OS can see that makes the SSD appear to be a traditional spinning drive. Once a read or write operation is commenced, the virtual drive circuits pass the operation through a translator in the SSD that maps the true inner guts of the SSD.

The *Non-Volatile Memory Express (NVMe)* specification supports a communication connection between the operating system and the SSD directly through a PCIe bus lane, reducing latency and taking full advantage of the wicked-fast speeds of high-end SSDs (see [Figure 8-16](#)). NVMe SSDs come in a couple of formats, such as an add-on expansion card, though most commonly in M.2 format. NVMe drives are more expensive than other SSDs, but offer much higher speeds. NVMe drives use SATAe.



**Figure 8-16** NVMe enables direct-to-the-bus communication.

## SCSI

SATA drives dominate the personal computer market, but another drive technology, called the *small computer system interface* (SCSI), rules the roost in the server market. SCSI has been around since the early days of HDDs and has evolved over the years from a parallel to a wider parallel to—and this should be obvious by now—a couple of super-fast serial interfaces. SCSI devices—parallel and serial—use a standard SCSI command set, meaning you can have systems with both old and new devices connected and they can communicate with no problem. SCSI drives used a variety of ribbon cables, depending on the version.

*Serial Attached SCSI* (SAS) hard drives provide fast and robust storage for servers and storage arrays today. The latest SAS interface, SAS-3, provides speeds of up to 12 Gbps. SAS controllers also support SATA drives, which is cool and offers a lot of flexibility for techs, especially in smaller server situations. SAS implementations offer literally more than a dozen different connector types. Most look like slightly chunkier versions of a SATA connector.

The CompTIA A+ certification includes SCSI, but surely that only means SAS. If you want to make the move to server tech, though, you'll definitely need to know about SCSI. The SCSI Trade Association (STA) Web site provides a good starting point: [www.scsita.org](http://www.scsita.org).

## Protecting Data with RAID

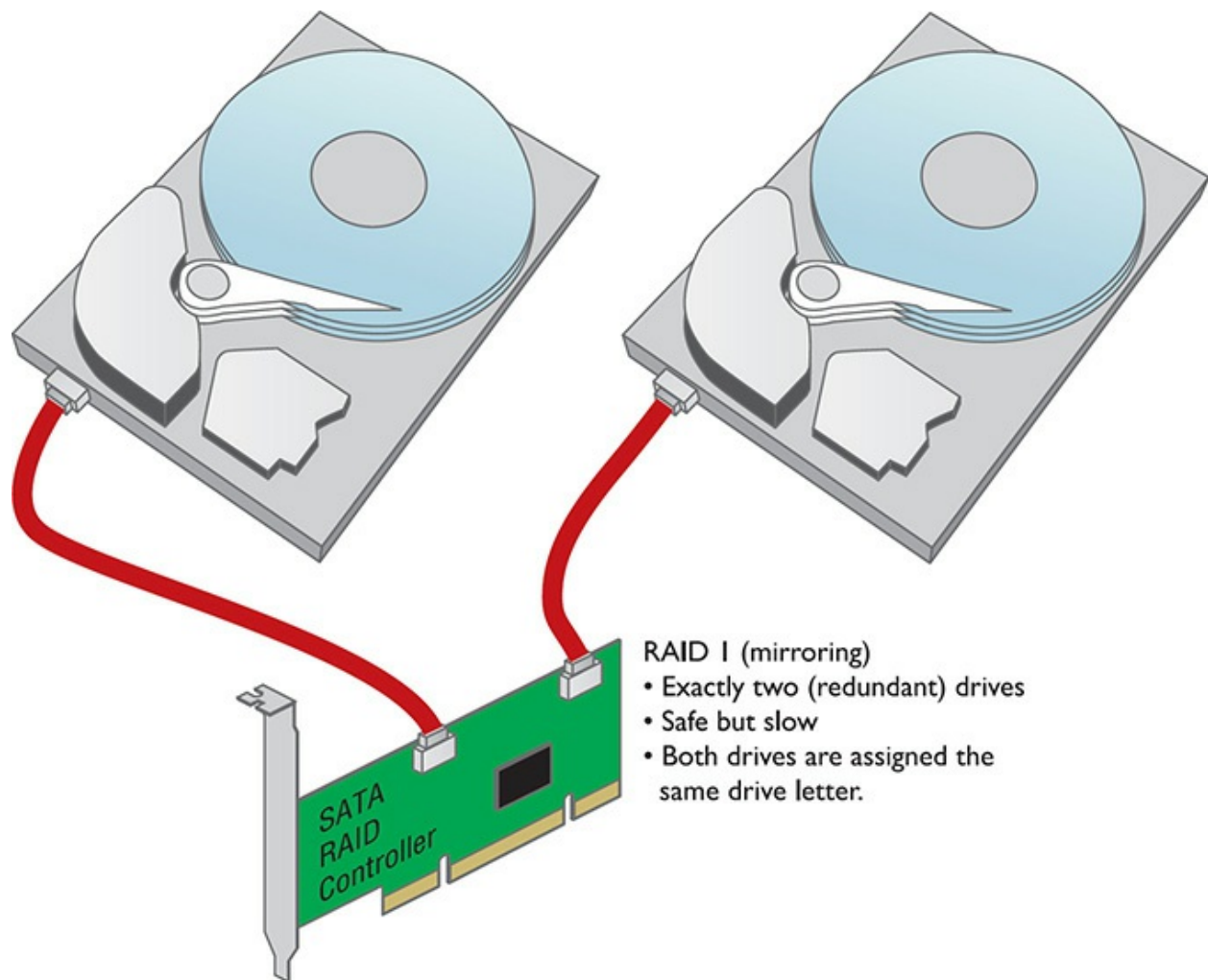
Ask experienced techs “What is the most expensive part of a PC?” and they'll all answer in the same way: “It's the data.” You can replace any single part of your PC for a few hundred dollars at most, but if you lose critical data—well, let's just say I know of two small companies that went out of business 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. 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, you could install some fancy hard drive controller that reads and writes data to two hard drives simultaneously (see [Figure 8-17](#)). 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*.

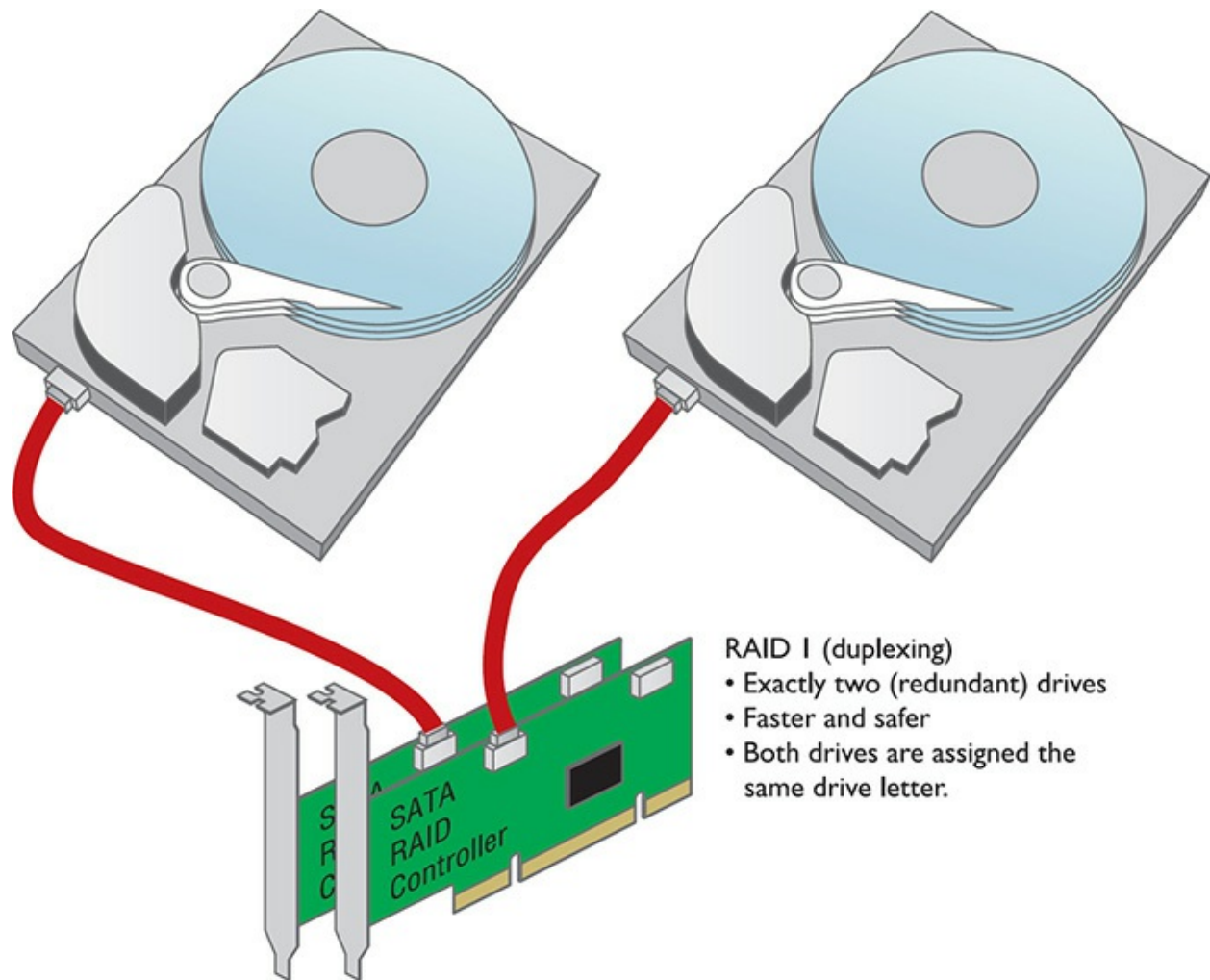


---

**Figure 8-17** Mirrored drives

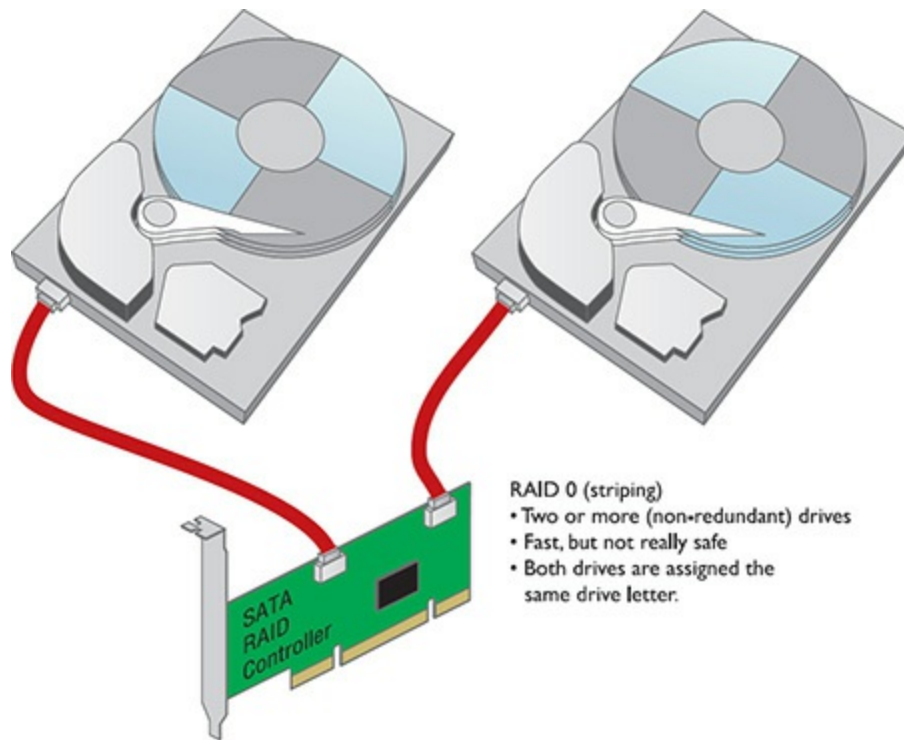
If you really want to make data safe, you can use a separate controller for each drive. With two drives, each on a separate controller, the system will

continue to operate even if the primary drive's controller stops working. This super-drive mirroring technique is called *disk duplexing* (see [Figure 8-18](#)). Disk duplexing is also marginally faster than disk mirroring because one controller does not write each piece of data twice.



**Figure 8-18** Duplexing drives

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 (see [Figure 8-19](#)).



---

**Figure 8-19** Disk striping

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. You should not do disk striping—unless you’re willing to increase the risk of losing data to increase the speed at which your hard drives store and retrieve data.



**NOTE** In practice (as opposed to benchmarking) you won’t experience any performance difference between mirroring and striping.

*Disk striping with parity*, in contrast, protects data by adding extra information, called *parity data*, that can be used to rebuild data if one of the drives fails. 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.

---



**NOTE** There is actually a term for a storage system composed of multiple independent disks of various sizes, *JBOD*, which stands for *just a bunch of disks* (or *drives*). Many drive controllers support JBOD.

## RAID

A couple of sharp guys in Berkeley back in the 1980s organized many of the techniques for using multiple drives for data protection and increasing speeds as the *redundant array of independent (or inexpensive) disks (RAID)*. An *array* describes two or more drives working as a unit. They outlined several forms or “levels” of RAID that have since been numbered 0 through 6 (plus a couple of special implementations). Only a few of these RAID types are in use today: 0, 1, 5, 6, 10, and 0+1.

- **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. I’ve heard this called *scary RAID* for that very reason.
- **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 because the data is duplicated; you need two 2-TB drives to store 2 TB of data.
- **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 requires at least three drives. RAID 5 arrays effectively use one drive’s worth of space for parity. If, for example, you have three 2-TB drives, your total storage capacity is 4 TB. If you have four 2-TB drives, your total capacity is 6 TB.





**NOTE** RAID 5 sounds great on paper and will seem great on your CompTIA A+ exam, but it's out of favor today. The failure rate of drives combined with the huge capacity (and rebuilding times) mean most RAID implementations shy away from the "lose only one drive" RAID 5.

- **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 four drives, but in exchange you can lose up to two drives at the same time.
- **RAID 10—Nested, Striped Mirrors** RAID levels have been combined to achieve multiple benefits, including speed, capacity, and reliability, but these benefits must be purchased at a cost, and that cost is efficiency. Take for instance RAID 10, also called RAID 1+0 and sometimes a "stripe of mirrors." Requiring a minimum of four drives, a pair of drives is configured as a mirror, and then the same is done to another pair to achieve a pair of RAID 1 arrays. The arrays look like single drives to the operating system or RAID controller. So now, with two drives, we can block stripe across the two mirrored pairs (RAID 0). Cool, huh? We get the speed of striping and the reliability of mirroring at the cost of installing two bytes of storage for every byte of data saved. Need more space? Add another mirrored pair to the striped arrays!
- **RAID 0+1—Nested, Mirrored Stripes** Like RAID 10, RAID 0+1 (or a "mirror of stripes") is a nested set of arrays that works in opposite configuration from RAID 10. It takes a minimum of four drives to implement RAID 0+1. Start with two RAID 0 striped arrays, then mirror the two arrays to each other. Which is better: the RAID 10 or the RAID 0+1? Why not do a bit of research and decide for yourself?



**EXAM TIP** In preparation for the CompTIA A+ 220-1001 exam, you'll want to be familiar with RAID levels 0, 1, 5, and 10. Know the minimum number of drives in a given level array, and how many failures a given array can withstand and remain functional.

RAID Level	Minimum Drives	Number of Functional Failures
RAID 0	2	0
RAID 1	2	1
RAID 5	3	1
RAID 6	4	2
RAID 10	4	Up to 2

## 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 methods can be used to set up RAID. The method you use depends largely on the level of RAID you desire, the operating system you use, and the thickness of your wallet.

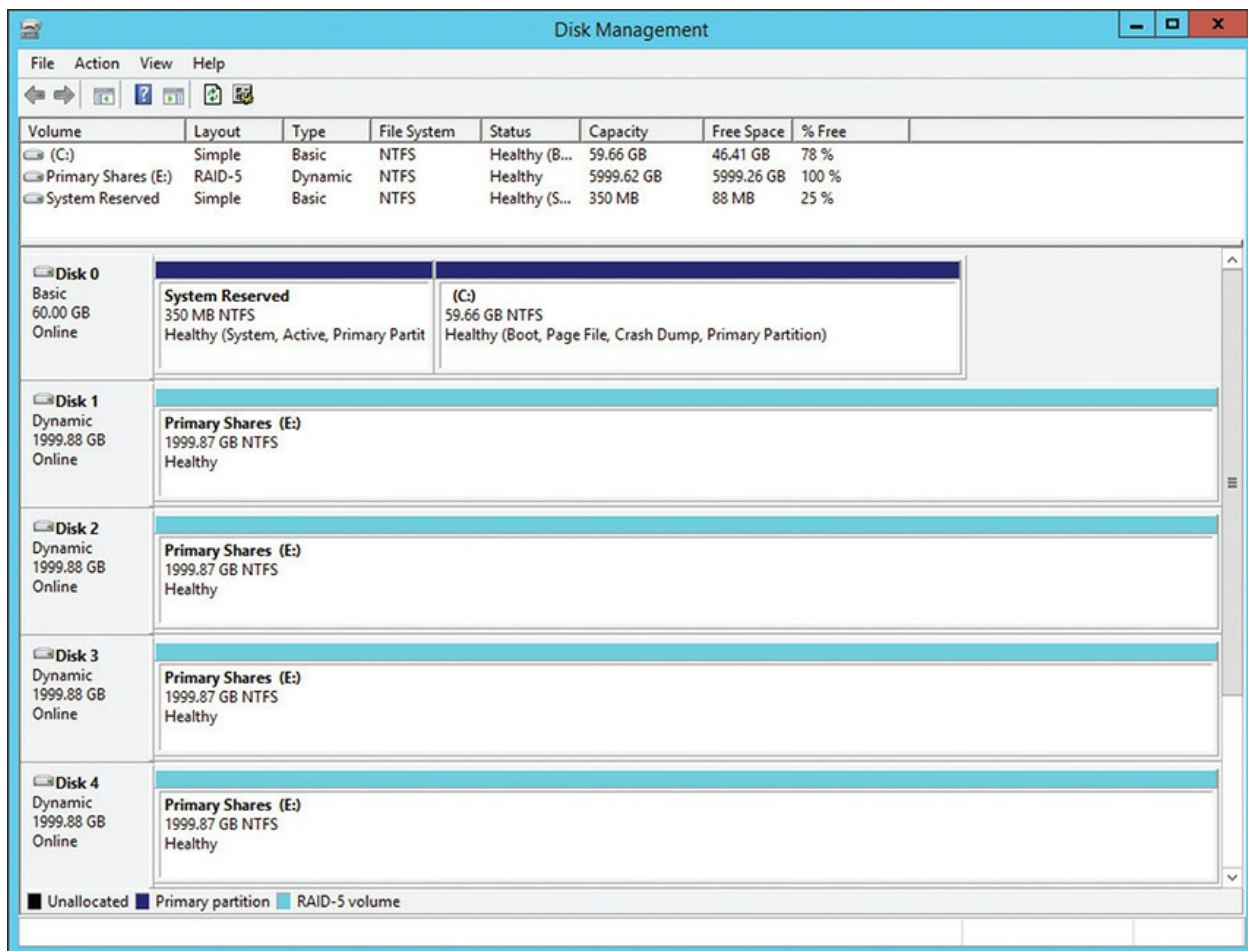
The obvious starting place for RAID is to connect at least two hard drives in some fashion to create a RAID array. Specialized RAID controller cards support RAID arrays of up to 15 drives—plenty to support even the most complex RAID needs. Dedicated storage boxes with built-in RAID make implementing a RAID solution simple for external storage and backups.

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

## Software Versus Hardware

All RAID implementations break down into either software or hardware 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 SATA controllers 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. The Disk Management program in Windows Server versions can configure drives for RAID 0, 1, or 5, and it works with PATA or SATA (see [Figure 8-20](#)). Windows 7/8/8.1/10 Disk Management can do RAID 0 and 1.



**Figure 8-20** Disk Management tool of Computer Management in Windows Server



**NOTE** [Chapter 9](#), “Implementing Mass Storage,” discusses RAID solutions implemented in Windows.

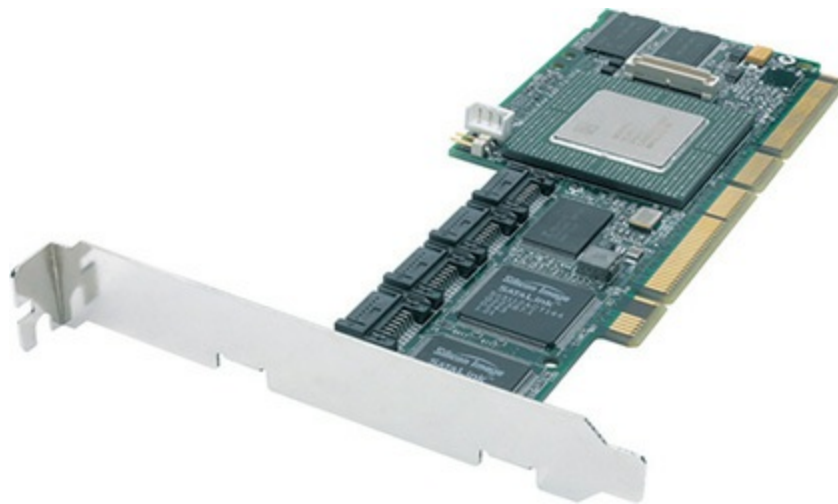
Windows Disk Management is not the only software RAID game in town. A number of third-party software programs work with Windows or other operating systems.

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.



**NOTE** See [Chapter 9](#) for a thorough discussion of *Storage Spaces*, a software RAID implementation available in Windows 8/8.1/10.

Hardware RAID centers on an *intelligent* controller that handles all of the RAID functions (see [Figure 8-21](#)). Unlike regular PATA/SATA controllers, these controllers have chips with their own processor and memory. This allows the card or dedicated box, instead of the operating system, to handle all of the work of implementing RAID.



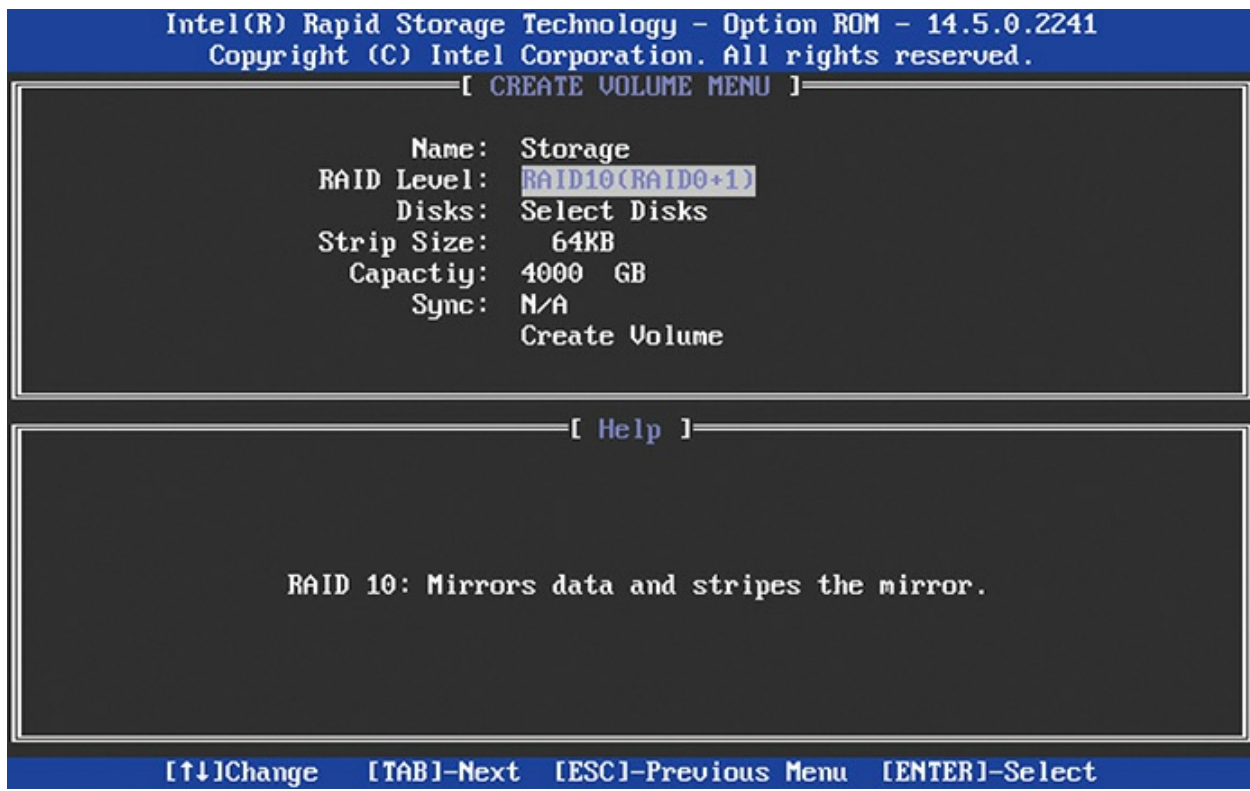
---

**Figure 8-21** Serial ATA RAID controller



Most traditional 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.

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 8-22 shows a typical firmware program used to configure a hardware RAID solution.



**Figure 8-22** RAID configuration utility



**SIM** Check out the [Chapter 8 Challenge! sim](#), “Storage Solution,” to examine best RAID practices at <http://totalsem.com/100x>.

## Dedicated RAID Boxes

Many people add a dedicated RAID box to add both more storage and a place to back up files. These devices take two or more drives and connect via one of the ports on a computer, such as USB or Thunderbolt (on modern systems) or FireWire or eSATA (on older systems). (See [Chapter 10](#) for details on USB and FireWire.) [Figure 8-23](#) shows an external RAID box (also called an *enclosure*). This model is typical, offering three options for the two drives inside: no RAID, RAID 0, or RAID 1.



---

**Figure 8-23** Western Digital RAID enclosure

## Installing Drives

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



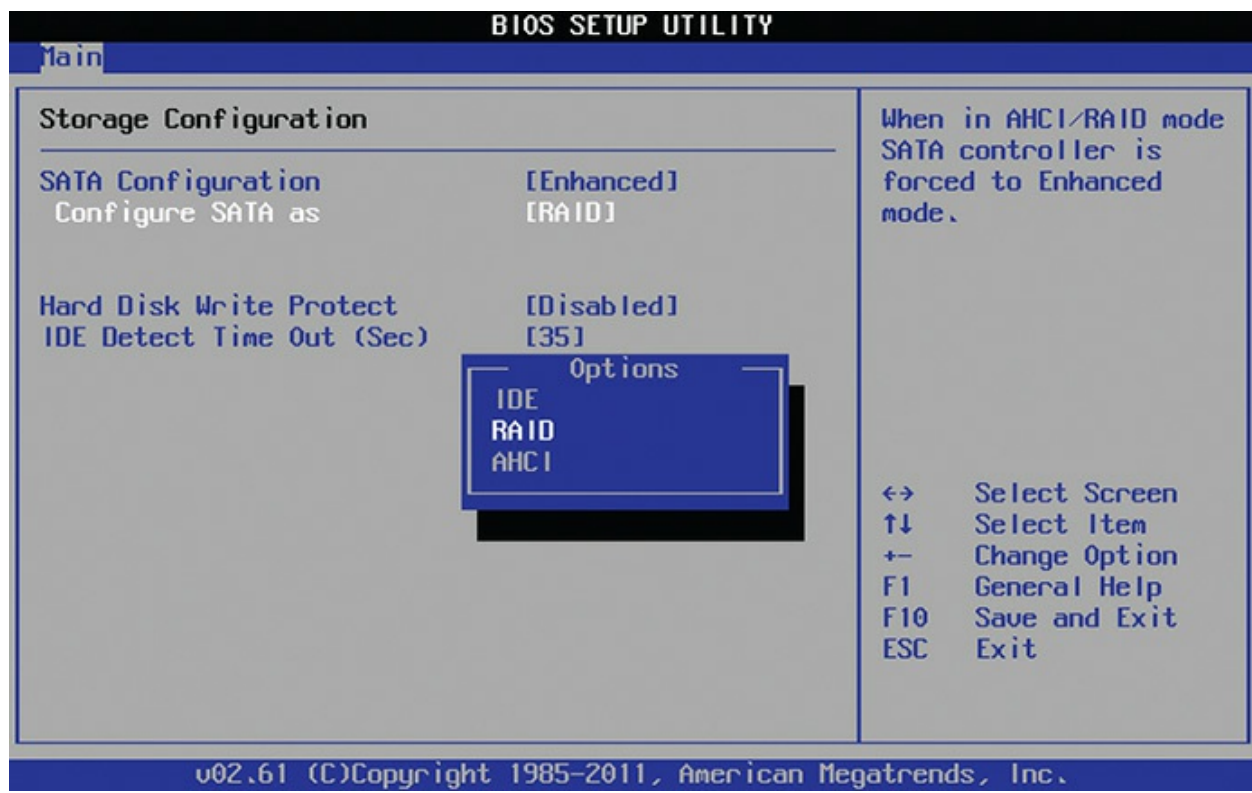
---

**EXAM TIP** Don't let the length of explanation about installation throw

you during CompTIA A+ 1001 exam prep. PATA installation is much more complicated than SATA installation, so we've devoted more ink to the process here. SATA is what you will most likely see in the field and on the exam.

## Choosing Your Drive

First, decide where you're going to put the drive. If you have a new motherboard, just slip the drive into the M.2 socket and secure it with the tiny screw. If you plan to install a 3.5-inch HDD or 2.5-inch SSD, then you need to go old school. Look for an open SATA connection. Is it part of a dedicated RAID controller? Many motherboards with built-in RAID controllers have a CMOS setting that enables you to turn the RAID controller on or off (see [Figure 8-24](#)).



**Figure 8-24** 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.

### **Try This! Managing Heat with Multiple Drives**

Adding three or more fast magnetic hard drives into a cramped PC case can be a recipe for disaster to the unwary tech. While the heat generated may not threaten the fabric of the time-space continuum, heat reduces the life expectancy of drives and computers. You have to manage the heat inside a RAID-enabled system because such systems usually have more than the typical quantity of drives found in desktop computers. The easiest way to do this is to add fans.

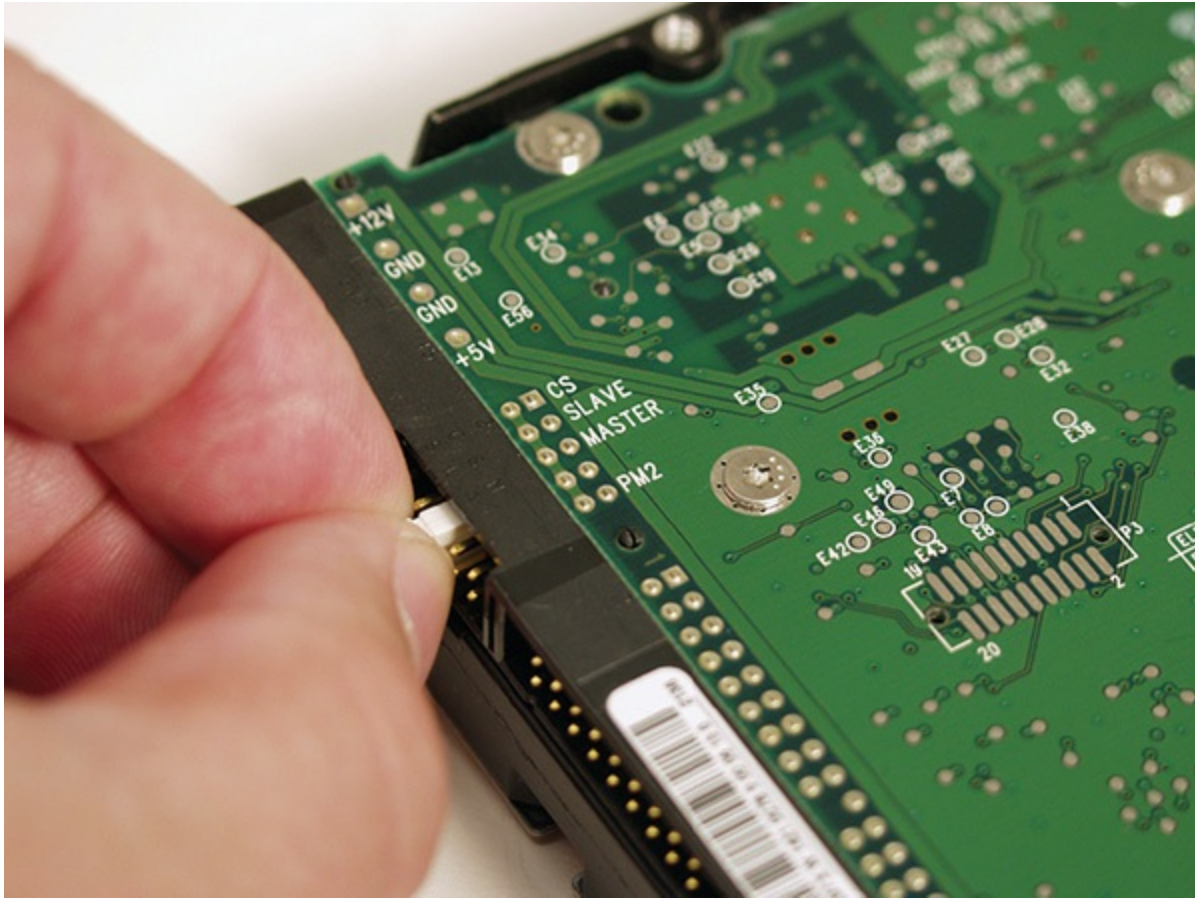
Open up the PC case and look for built-in places to mount fans. How many case fans do you have installed now? What size are they? What sizes can you use? (Most cases use 80-mm fans, but 120-mm and even larger fans are common as well.) Jot down the fan locations of the case and take a trip to the local PC store or online retailer 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 achieve a compromise between keeping your case cool enough and avoiding early deafness.

## **PATA Drive Installation**

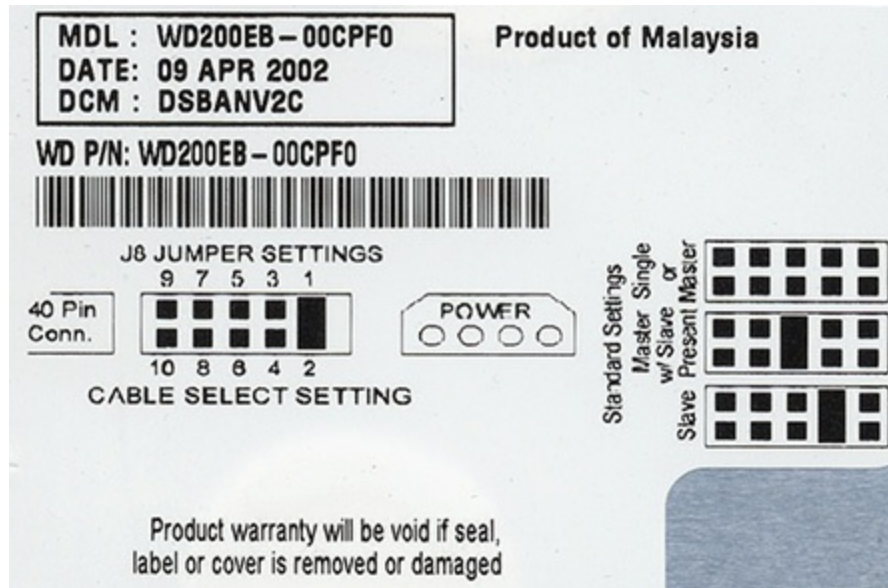
Sorry, but CompTIA still has PATA (IDE) drives obliquely listed as a competency, so let's go through installation of these ancient drives quickly. PATA drives have jumpers on the drive that must be set properly. If you have only one hard drive, set the drive's jumpers to master or standalone. If you have two drives, set one to master and the other to slave. Or set both to cable select. See [Figure 8-25](#) for a close-up of a PATA hard drive, showing the jumpers.





**Figure 8-25** Master/slave jumpers on a PATA drive

Some drives don't label the jumpers *master* and *slave*. So how do you know how to set them properly? The easiest way is to read the front of the drive; you'll find a diagram on the housing that explains how to set the jumpers properly. [Figure 8-26](#) shows the label of one of these drives, so you can see how to set the drive to master, slave, or cable select.



**Figure 8-26** Drive label showing master/slave settings

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.

Older motherboards have dedicated PATA ports built in, so you connect PATA cables directly to the motherboard. Newer motherboards do not have such ports, so to connect a PATA drive requires a special add-on PATA controller expansion card.

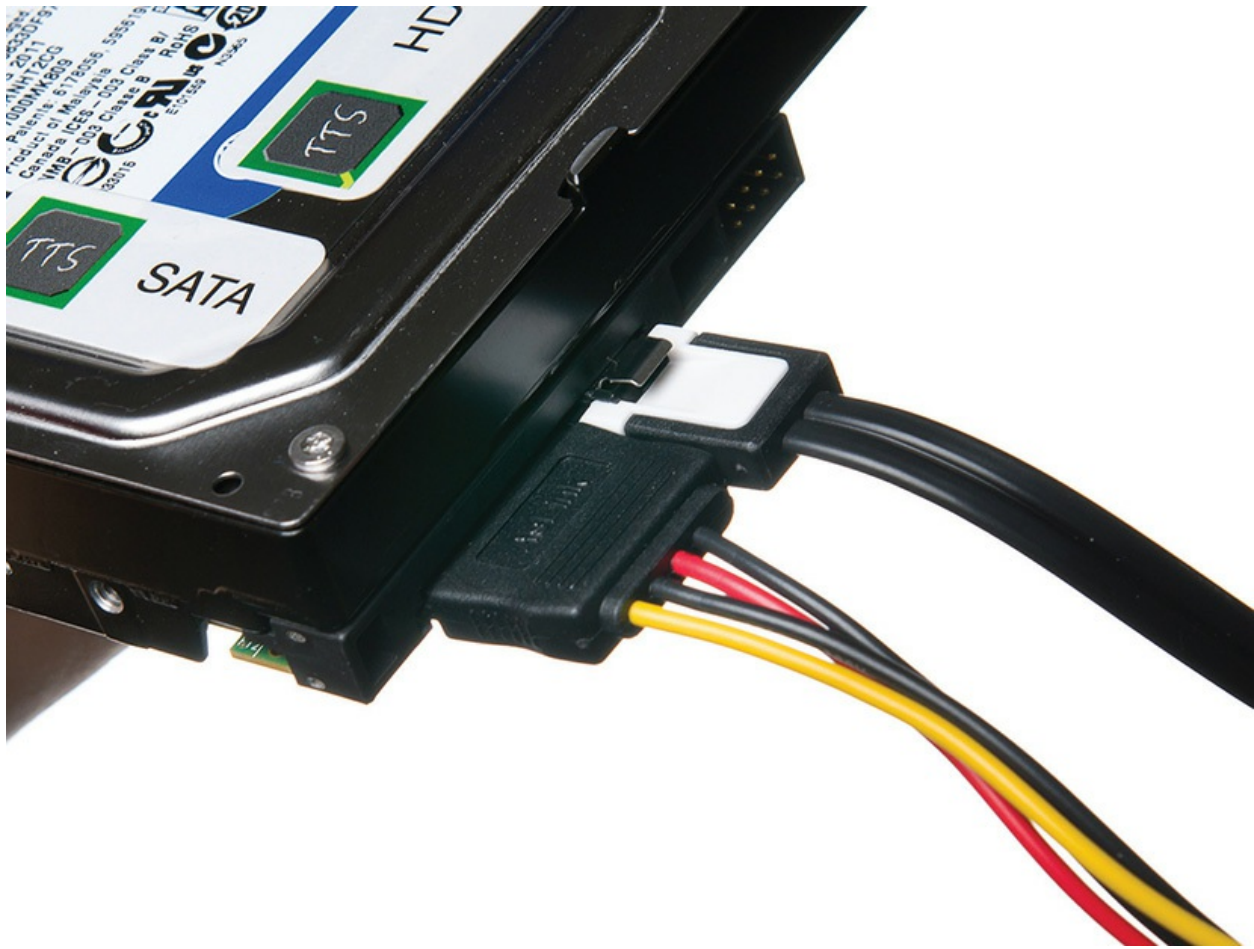
Finally, you need to plug a Molex connector from the power supply into the drive. All PATA drives use a Molex connector. Okay, that's it—no more PATA discussion!



**EXAM TIP** The CompTIA A+ 1001 objectives list PATA motherboard ports as *IDE connectors*. Don't get thrown off by the different terminology!

## Cabling SATA Drives

Installing SATA hard disk drives is much easier than installing PATA devices because 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 8-27](#)—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 8-27** Properly connected SATA cable



**NOTE** Some older SATA drives have jumpers, but they are used to

configure SATA version/speed (1.5, 3.0) or power management. The rule of one drive for one controller applies to these drives, just like more typical jumperless SATA drives.

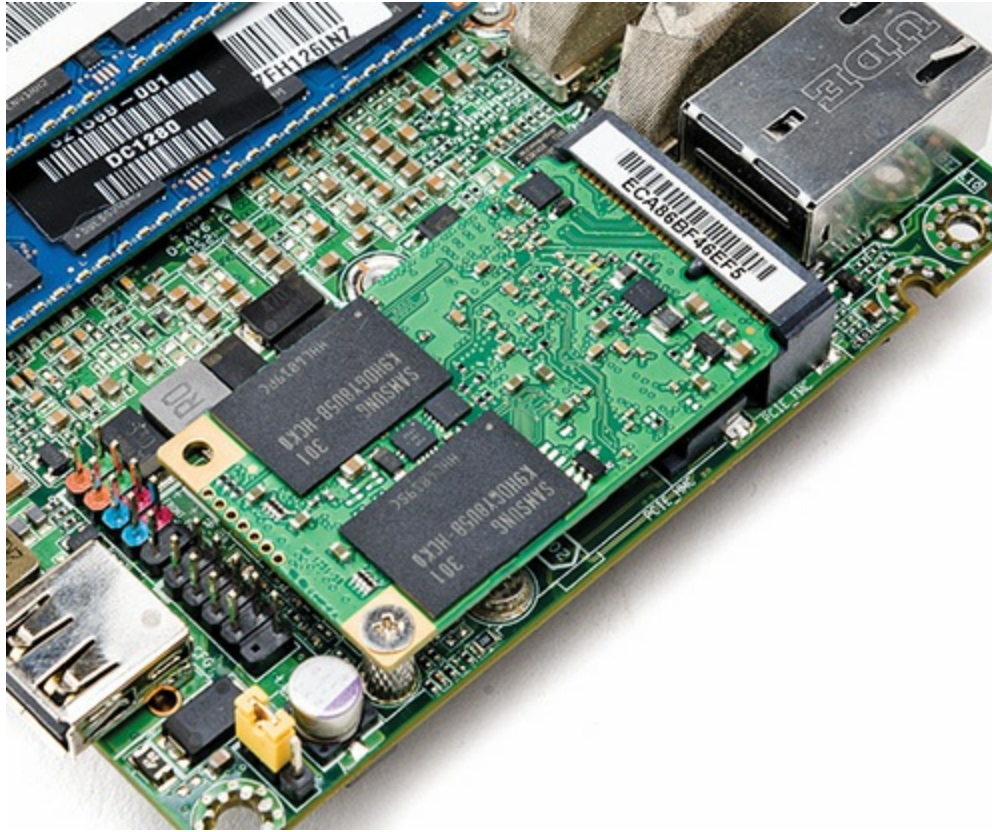
Every modern motherboard has two or more SATA ports (or *SATA connectors*) built in, like you saw in pictures in [Chapter 6](#), “Motherboards.” The ports are labeled (SATA 1 to however many are included). Typically, you install the primary drive into SATA 1, the next into SATA 2, and so on. With non-booting SATA drives, such as in M.2 motherboards, it doesn’t matter which port you connect the drive to.

## **Connecting Solid-State Drives**

SATA SSDs possess the same connectors as magnetic SATA drives, so you install an SSD as you would any SATA drive. SATA SSDs usually come in 2.5-inch laptop sizes. Just as with earlier hard drive types, you either connect SSDs correctly and they work, or you forget to plug in the power cable and they don’t.

M.2 and mSATA drives slip into their slot on the motherboard or add-on card, then either clip in place or secure with a tiny screw (see [Figure 8-28](#)). Both standards are keyed, so you can’t install them incorrectly.





**Figure 8-28** mSATA SSD secured on motherboard

Keep in mind the following considerations before installing or replacing an existing HDD with an SSD:

- Do you have the appropriate drivers and firmware for the SSD? Newer Windows versions will load the most currently implemented SSD drivers. As always, check the manufacturer's specifications as well.
- Do you have everything important backed up? Good!

## **BIOS Support: Configuring CMOS and Installing Drivers**

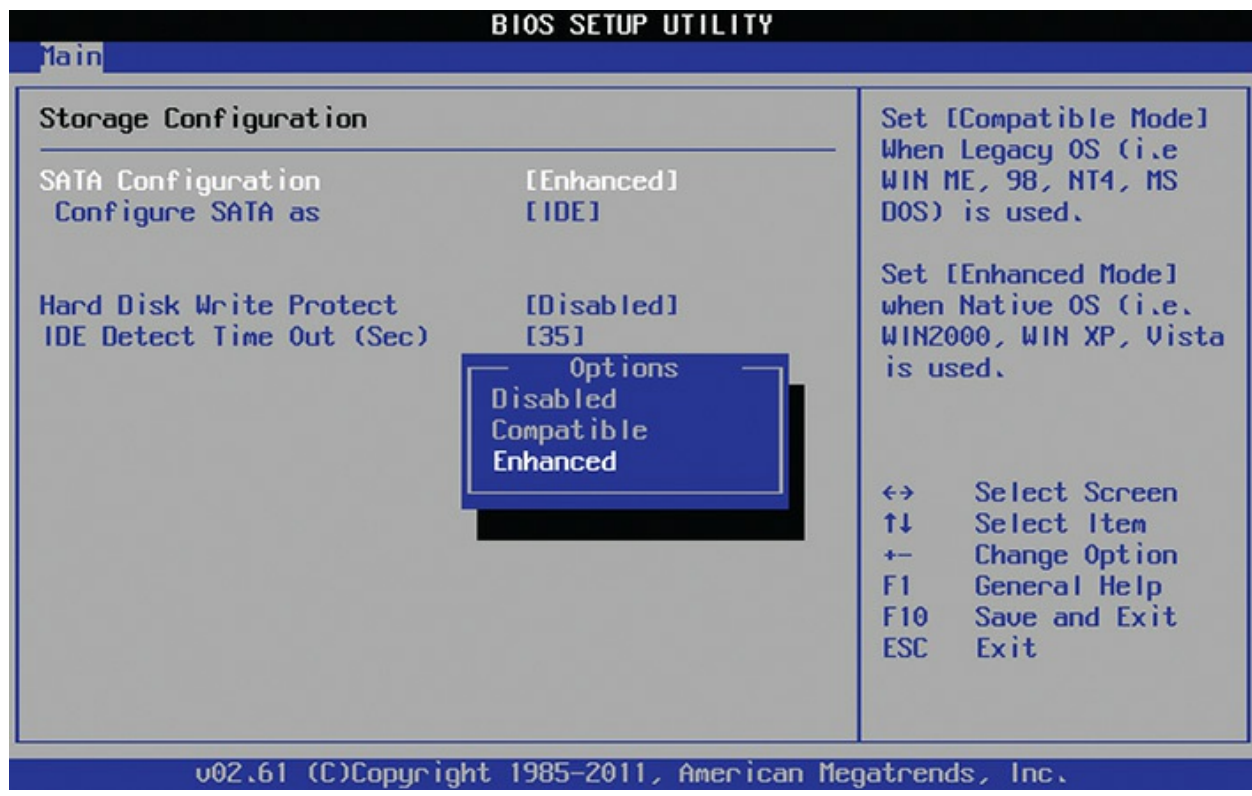
Every device in your PC needs BIOS support, whether it's traditional BIOS or UEFI. Hard drive controllers are no exception. Motherboards provide support for the SATA hard drive controllers via the system BIOS, but they require configuration in CMOS for the specific hard drives attached.

In the old days, you had to fire up CMOS and manually enter hard drive

information whenever you installed a new drive. Today, this process is automated.

## Configuring Controllers

As a first step in configuring controllers, make certain they're enabled. Most controllers remain active, ready to automatically detect new drives, but you can disable them. Scan through your CMOS settings to locate the controller on/off options (see [Figure 8-29](#) for typical settings). This is also the time to check whether the onboard RAID controllers work in both RAID and non-RAID settings.



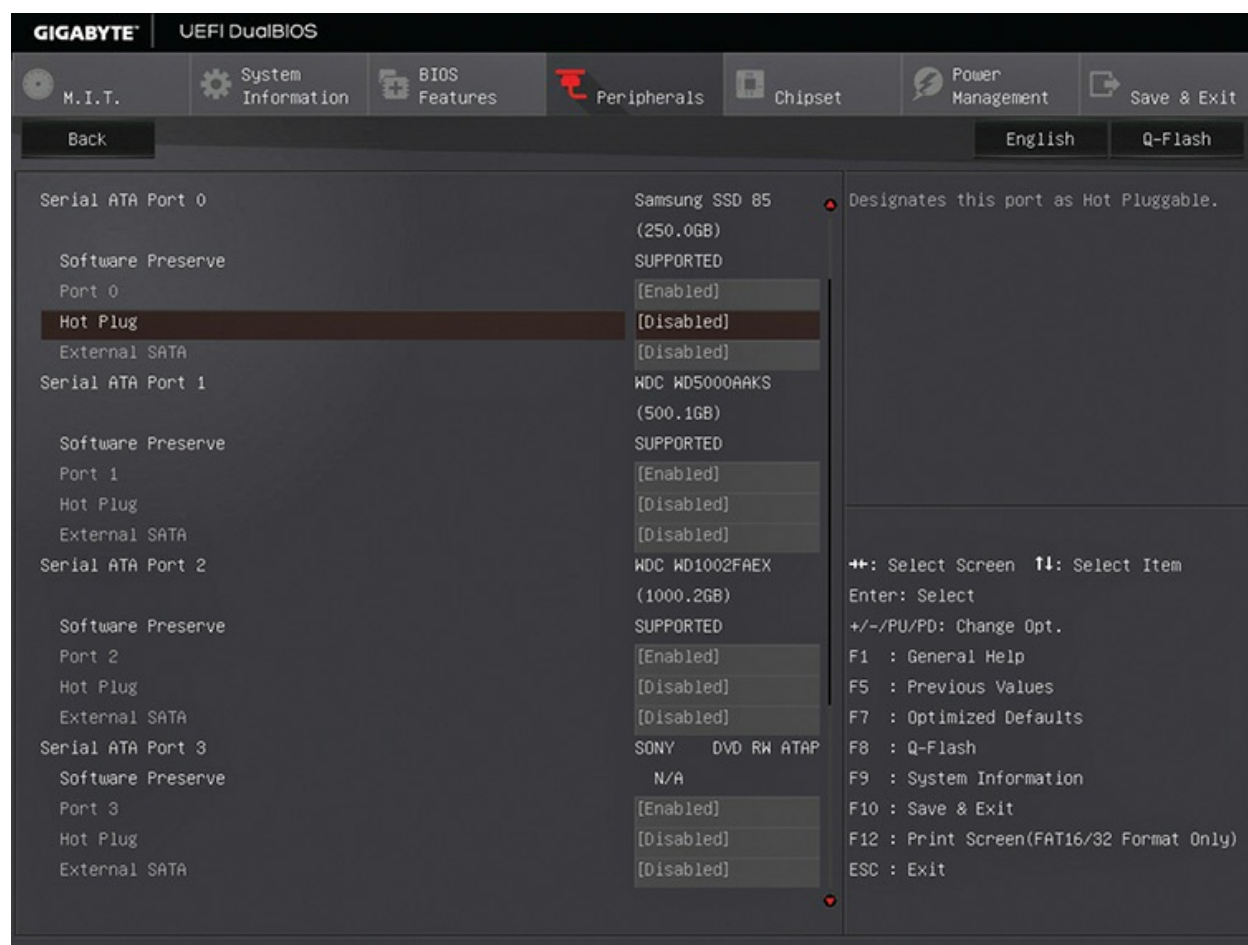
**Figure 8-29** Typical controller settings in CMOS

## 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 that takes almost all the work out of configuring hard drives. Motherboards use a numbering system to

determine how drives are listed—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. So instead of names of drives, you see numbers. Look at [Figure 8-30](#).



**Figure 8-30** Standard CMOS features

Whew! Lots of hard drives! This motherboard supports six SATA connections. Each connection has a number, with an M.2 SSD on SATA 0, hard drives on SATA 1 and SATA 2, and the optical drive on SATA 3. Each was autodetected and configured by the BIOS without any input from me. Oh, to live in the future!

## Boot Order

If you want your computer to run, it's going to need an operating system to boot. You assign *boot order* priority to drives and devices in CMOS.

Figure 8-31 shows a typical boot-order screen, with a first, second, and third boot option. Many users like to boot first from the optical drive and then from a hard drive. This enables them to put in a bootable optical disc 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 8-31** Boot order

Most modern CMOS setup utilities include a second screen for determining the boot order of your hard drives. You might want to set up a boot order that goes optical drive, followed by hard drive, and then USB thumb drive, but what if you have more than one hard drive? This screen enables you to set which hard drive goes first. If you have a different



operating system on each hard drive, this can be very helpful.

## **Enabling AHCI**

On motherboards that support AHCI, you implement it in CMOS. You'll generally have up to three options/modes/HBA configurations: IDE/SATA or compatibility mode, AHCI, or RAID. Don't install modern operating systems in compatibility mode; it's included with some motherboards to support ancient (Windows XP) or odd (some Linux distros, perhaps?) operating systems. AHCI works best for current HDDs and SSDs, so make sure the HBA configuration is set to AHCI.

## **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 most obvious question, especially during installation, is "Did I plug it in correctly? Or did I plug both data and power in correctly?" With autodetection, the answer is simple: If the system doesn't see the drive, 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 improperly, or messed up some other connectivity issue. To troubleshoot hard drives, simply work your way through each step to figure out what went wrong.

Make sure the BIOS recognizes the hard drive. Use the CMOS setup program to check. Check the physical connections, then run through these issues in CMOS. Is the controller enabled? Similarly, can the motherboard support the type of drive you're installing? If not, you have a couple of options. You may be able to flash the BIOS with an upgraded BIOS from the manufacturer or get a hard drive controller that goes into an expansion slot.

## **Chapter Review**

### **Questions**

1. Which of the following is a common spindle speed for an HDD?



- A. 5200
  - B. 7200
  - C. 9200
  - D. Not applicable. HDDs have no moving parts.
2. Which form factor connects directly to a dedicated motherboard socket?
- A. 2.5-inch SSD
  - B. 3.5-inch SSD
  - C. M.2 SSD
  - D. eSATA SSD
3. How many PATA hard drives can you have on a system with two PATA hard drive controllers?
- A. 1
  - B. 2
  - C. 3
  - D. 4
4. How do you differentiate two PATA drives on the same cable?
- A. The flat ribbon cable has a seven-wire twist that determines which is which.
  - B. You set jumpers on the individual drives to determine which is master and which is slave.
  - C. The PATA controller determines the hierarchy.
  - D. Both drives are considered equal.
5. What happens if you cable a PATA 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. The drive installs in compatibility mode.
6. What is the maximum cable length of an internal SATA device?
- A. 2 meters

- B.** 12 inches
  - C.** 18 inches
  - D.** 1 meter
- 7.** 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/host card.
- 8.** Which SATA version offers the least overhead (and thus best performance)?
  - A.** AHCI
  - B.** SATA 2.0
  - C.** PATA 3.0
  - D.** SATAe
- 9.** Which standard supports magnetic SATA drives most efficiently?
  - A.** AHCI
  - B.** CMOS
  - C.** SATA-IO
  - D.** SATA 3.2
- 10.** Which RAID standard requires at least four drives?
  - A.** RAID 1
  - B.** RAID 4
  - C.** RAID 5
  - D.** RAID 10

## Answers

- 1. B.** Common spindle speeds on magnetic hard drives are 5400, 7200, 10,000, and 15,000 RPM.

2. **C.** The M.2 (and mSATA) SSD has a dedicated motherboard socket.
3. **D.** Each controller supports two drives.
4. **B.** PATA drives use master/slave jumpers to differentiate between the two drives.
5. **C.** Nothing will be damaged or lost—there just won't be any communication.
6. **D.** The maximum cable length of an internal SATA device is 1 meter.
7. **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.
8. **D.** SATA Express (SATAe) uses the PCIe bus and has none of the traditional SATA overhead.
9. **A.** The AHCI standard supports magnetic SATA drives efficiently.
10. **D.** RAID 10 requires at least four drives.