CHAPTER 3

The Visible Computer

In this chapter, you will learn how to

- Describe how computing devices work
- Identify common connectors and devices on a typical computer system
- Discuss features common to operating system software

Charles Babbage didn't set out to change the world. He just wanted to do math without worrying about human error, something all too common in his day. Babbage was a mathematician in the nineteenth century, a time well before anyone thought to create electronic calculators or computers (see Figure 3-1). When he worked on complex math, the best "computers" were people who computed by hand. They solved equations using pen and paper.

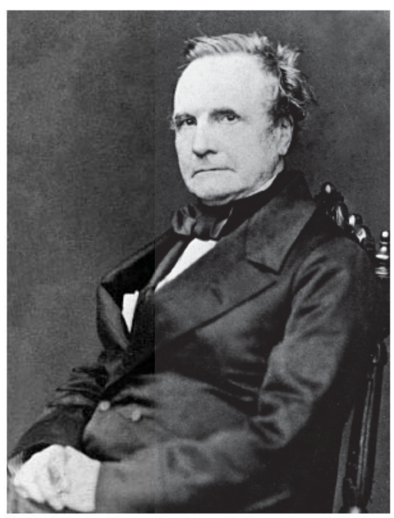


Figure 3-1 Charles Babbage, father of the computer

Babbage thought of making machines that would do calculations mechanically, so the numbers would always be right. Although his ideas were ahead of his time, inventors in the mid-twentieth century picked up the concepts and created huge calculating machines that they called *computers*.

This chapter explores how computing devices work. We'll look first at the computing process, then turn to hardware components common to all devices. The chapter finishes with a discussion about software, exploring commonality among all operating systems and specific functions of application programming. And, there are lots of pictures.

Historical/Conceptual

The Computing Process

In modern terms, a *computer* is an electronic device that can perform calculations. The most common types use special programming languages that people, known as *computer programmers*, have written and compiled to accomplish specific tasks.

When most people hear the word "computer," they picture *general* computing devices, machines that can do all sorts of things. The typical *personal computer (PC)* runs the operating system Microsoft Windows and is used for various tasks (see Figure 3-2). You can use it to manage your money and play games, for example, without doing anything special to it, such as adding new hardware.

Figure 3-2
A typical PC

Here are some other general-purpose computing devices:

- Apple Mac
- Apple iPad
- Smartphone
- Portable computer (see Figure 3-3)

Figure 3-3
A portable computer

Plenty of other devices do *specific* computing jobs, focusing on a single task or set of similar tasks. You probably encounter them all the time. Here's a list of common specific-purpose computers:

- Apple iPod
- Pocket calculator
- Digital watch
- Digital clock
- Wi-Fi picture frame
- Basic mobile phone
- Xbox One
- PlayStation 4
- GPS (Global Positioning System, the device that helps drivers figure out how to get where they need to go)
- TiVo
- Point of sale (POS) system (see Figure 3-4)
- Digital camera
- Camcorder

Figure 3-4
A point of sale computer in a gasoline pump

This list isn't even close to complete! Plus, there are computers *inside* a zillion other devices. Here are some:

- Modern refrigerators
- Every automobile built since 1995
- Airplanes

- Boats
- Mall lighting systems
- Zambonis
- Home security alarms

You get the idea. Computers help the modern world function.

NOTE I picked 1995 as an arbitrary date for when every new car built had a computer. Computers have been used with cars for a long time. Simple computers helped make car factories work better starting in the 1970s, for example. The earliest mass-production car I found that had a central processor chip for added performance was the BMW 3 Series. The 1985-86 BMW 325, for example, can gain a few extra horsepower just from a ~$200 chip upgrade.

Modern computer techs need to know how different types of computing devices work so they can support the many devices used by their clients. This diversity is also reflected in the CompTIA A+ exams.

If the list of devices to support seems overwhelming, relax. The secret savior for modern techs is that computing devices function similarly to each other. Once you know what a particular device should enable a user to do, you'll be able to configure and troubleshoot successfully.

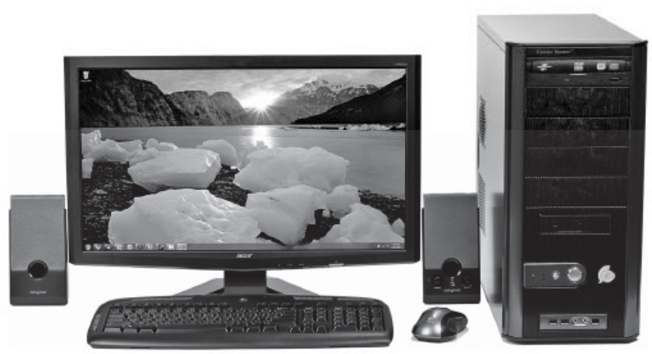
The Computing Parts

A modern computer consists of three major components:

- Hardware
- Operating system
- Applications

The *hardware* is the physical stuff that you can touch or hold in your hand. With a smartphone, for example, you hold the phone. On a typical personal computer, you touch the keyboard or view images on the monitor (see Figure 3-5).

Figure 3-5 A typical computer



The *operating system (OS)* controls the hardware and enables you to tell the computer what to do. The operating system often appears as a collection of windows and little icons you can click or touch (see Figure 3-6). Collectively these are called the *user interface (UI),* which means the software parts with which you can interact. The UI that offers images or icons to select (as opposed to making you type commands) is called a *graphical user interface (GUI).*

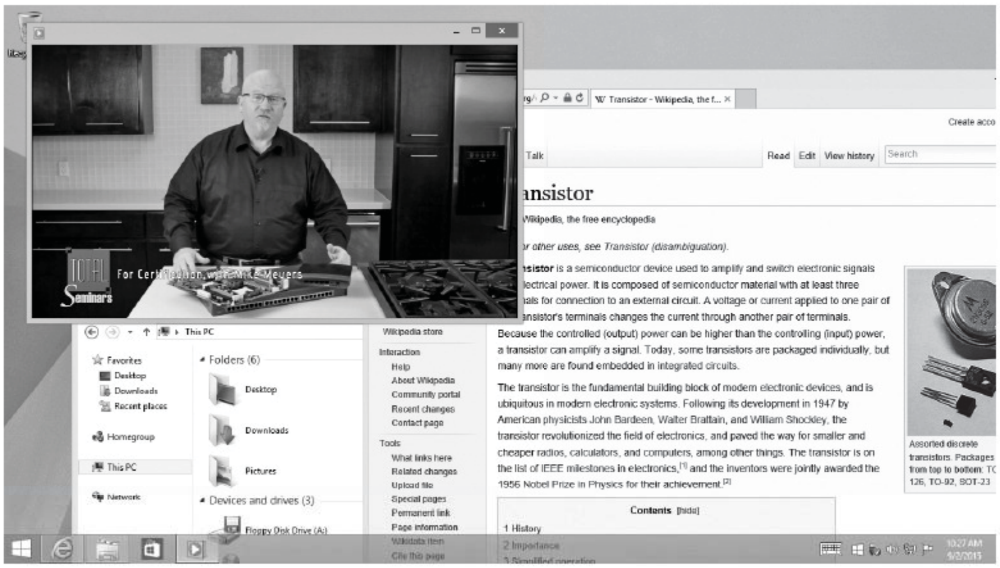


Figure 3-6 The Microsoft Windows 8.1 operating system

Applications (or programs) enable you to do specialized tasks on a computer, such as

- Type a letter
- Send a message from your computer in Houston to your friend's computer in Paris
- Wander through imaginary worlds with people all over Earth

Very simple computing devices might have an operating system with only a few features that give you choices. A digital camera, for example, has a menu system that enables you to control things like the quality of the picture taken (see Figure 3-7).

More complicated devices offer more choices. An Apple iPhone, for example, can do some cool things right out of the box, including make a phone call. But you can visit the Apple online store—the App Store—for programs and download applications (known as apps) to do all sorts of things that Apple didn't include (see Figure 3-8).

Finally, multipurpose computers like the typical Windows PC or Mac OS X computer offer applications to help you do everything from write a book on CompTIA A+ certification to talk with someone on the other side of the world, with full audio and video (see Figure 3-9).

Figure 3-7 Changing settings on a digital camera

Figure 3-8 Talking Carl + talks back to you—perhaps not the most *useful* app on the planet, but amusing

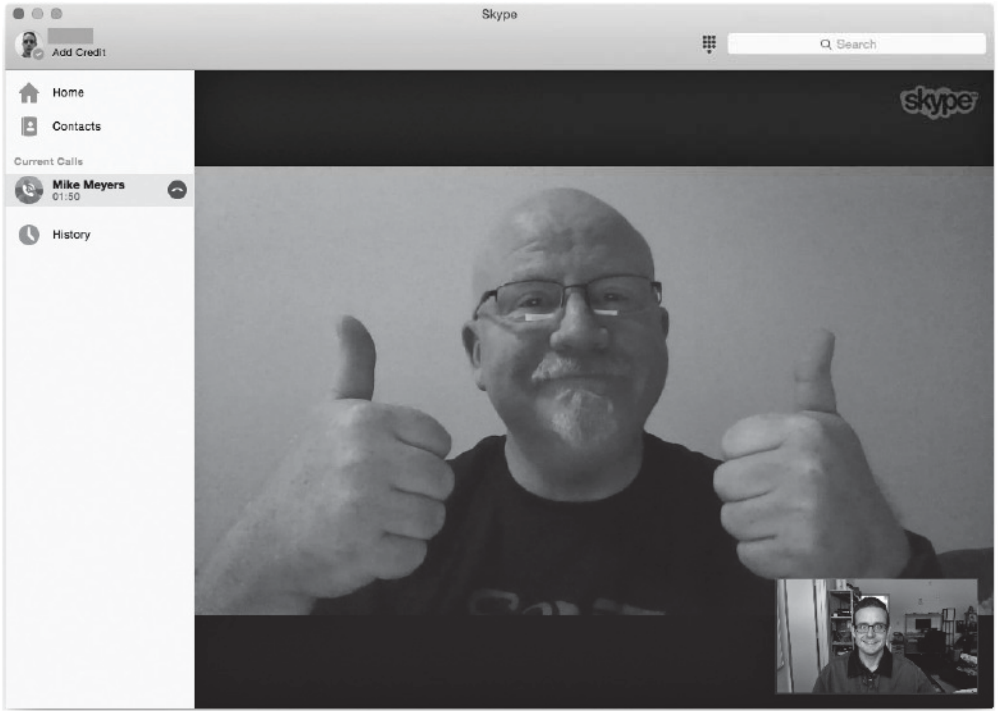


Figure 3-9 Skype communication

Stages

At the most basic level, computers work through three stages, what's called the *computing process*:

- Input
- Processing
- Output

You start the action by doing something—clicking the mouse, typing on the keyboard, or touching the touch screen. This is *input.* The parts inside the device or case take over at that point as the operating system tells the hardware to do what you've requested. This is *processing*.

In fact, at the heart of every computing device is a *central processing unit (CPU)*, usually a single, thin wafer of silicon and tiny transistors (see Figure 3-10). The CPU handles the majority of the processing tasks and is, in a way, the "brain" of the computer.

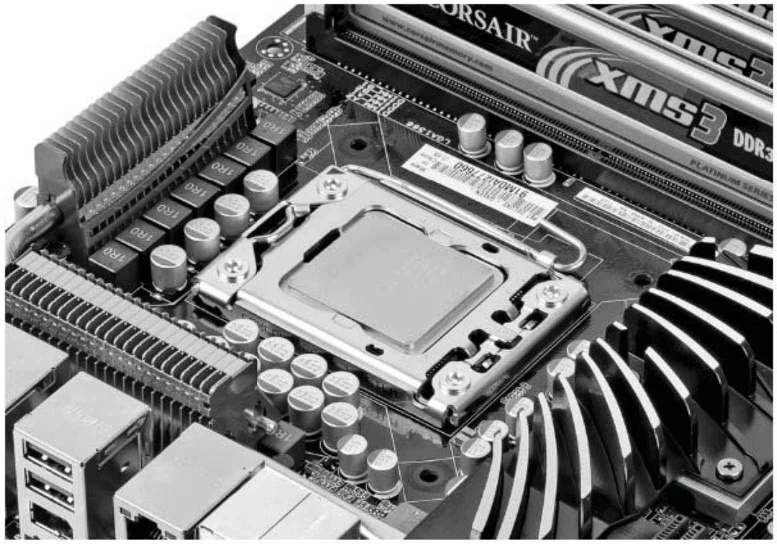


Figure 3-10 Intel Core i7 CPU on a motherboard

NOTE Chapter 4, "Microprocessors," gives a lot more information on CPUs and other processing components.

Once the computer has processed your request, it shows you the result by changing what you see on the display or playing a sound through the speakers. This is *output*. A computer wouldn't be worth much if it couldn't demonstrate that it fulfilled your commands! Figure 3-11 shows the computing process.

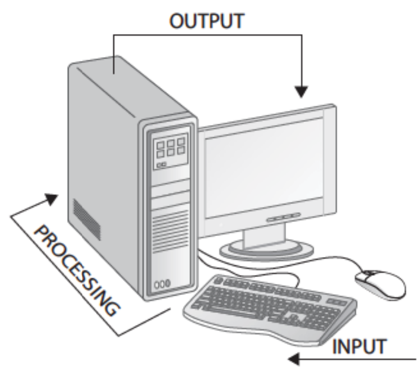


Figure 3-11 The computing process

Modern computing devices almost always have two other stages:

- Data storage
- Network connection

Data storage means saving a permanent copy of your work so that you can come back to it later. It works like this. First, you tell the computer to save something. Second, the CPU processes that command and stores the data. Third, the computer shows you something, such as a message saying that the data is stored. Any work that you *don't* save is lost when you turn the computer off or exit the application.

Most computing devices connect to other devices to access other resources. A *network connection* often describes how one computer connects to one or more other computers. And it doesn't just apply to a couple of office computers. Every smartphone, for example, can connect to the Internet and play a video from YouTube (assuming you have a signal from a cell tower and a data plan). A network connection can also mean running a cable between two devices, like connecting an iPad or iPhone to a Windows desktop machine using a Lightning-to-USB cable.

At this point, students often ask me a fundamental question: "Why should I care about the computing process?" The answer to this question defines what makes a good computer technician. Here's my response.

Why the Process Matters to Techs

Because the computing process applies to every computing device, it provides the basis for how every tech builds, upgrades, and repairs such devices. By understanding both the components involved and how they talk to each other, you can work with *any* computing device. It might take a couple minutes to figure out how to communicate with the device via input, for example, but you'll quickly master it because you know how all computing devices work.

Breaking It Down

The whole computer process from start to finish has a lot of steps and pieces that interact. The more you understand about this interaction and these pieces, the better you can troubleshoot when something goes wrong. *This is the core rule to being a great tech.*

Here are nine steps that apply to most computers and computing devices when you want to get something done:

1. Power up. Computers run on electricity.
2. Processing parts prepare for action.
3. You provide input.
4. Processing parts process your command.
5. Processing parts send output information to your output devices.
6. Output devices show you the results.
7. Repeat Steps 3–6 until you're satisfied with the outcome.
8. Save your work.
9. Power down the computer.

We'll come back to these processing steps as we tackle troubleshooting scenarios throughout the book. Keep these steps in mind to answer the essential question a tech should ask when facing a problem: What can it be? Or, in slightly longer fashion: What could cause the problem that stopped this device from functioning properly?

901

Computing Hardware

A lot of this book takes you in depth on specific computing hardware, such as CPUs and mass storage devices. CompTIA expects competent techs to know what to call every connector, socket, and slot in a variety of computing devices. Rather than describe all of those briefly here, I decided to create a photo walkthrough naming points-of-interest and the chapters that discuss them.

EXAM TIP Memorize the names of the components, connectors, and terms discussed and displayed in this section. You'll see them in future chapters, in the real world, and on the CompTIA A+ 901 exam.

This section serves as a visual introduction to the components and connections. Plus, it should work great as a set of study sheets for memorizing names just before taking the 901 exam. The images that follow indicate the chapters where you'll find information about a component or connection standard.

Figure 3-12 shows a typical PC. The input and output devices should be familiar to most.

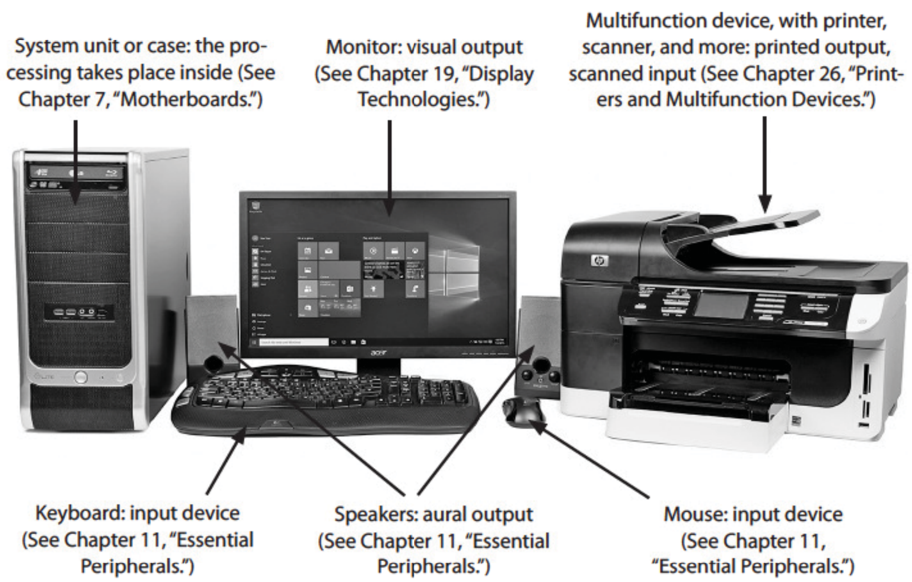


Figure 3-12 PC with common peripherals

Figure 3-13 shows the back of a PC's system unit, where you'll find the many connection points called ports. Some ports connect to output devices; a couple are exclusively used for input devices. Most (such as the universal serial bus, or USB) handle either type of device.

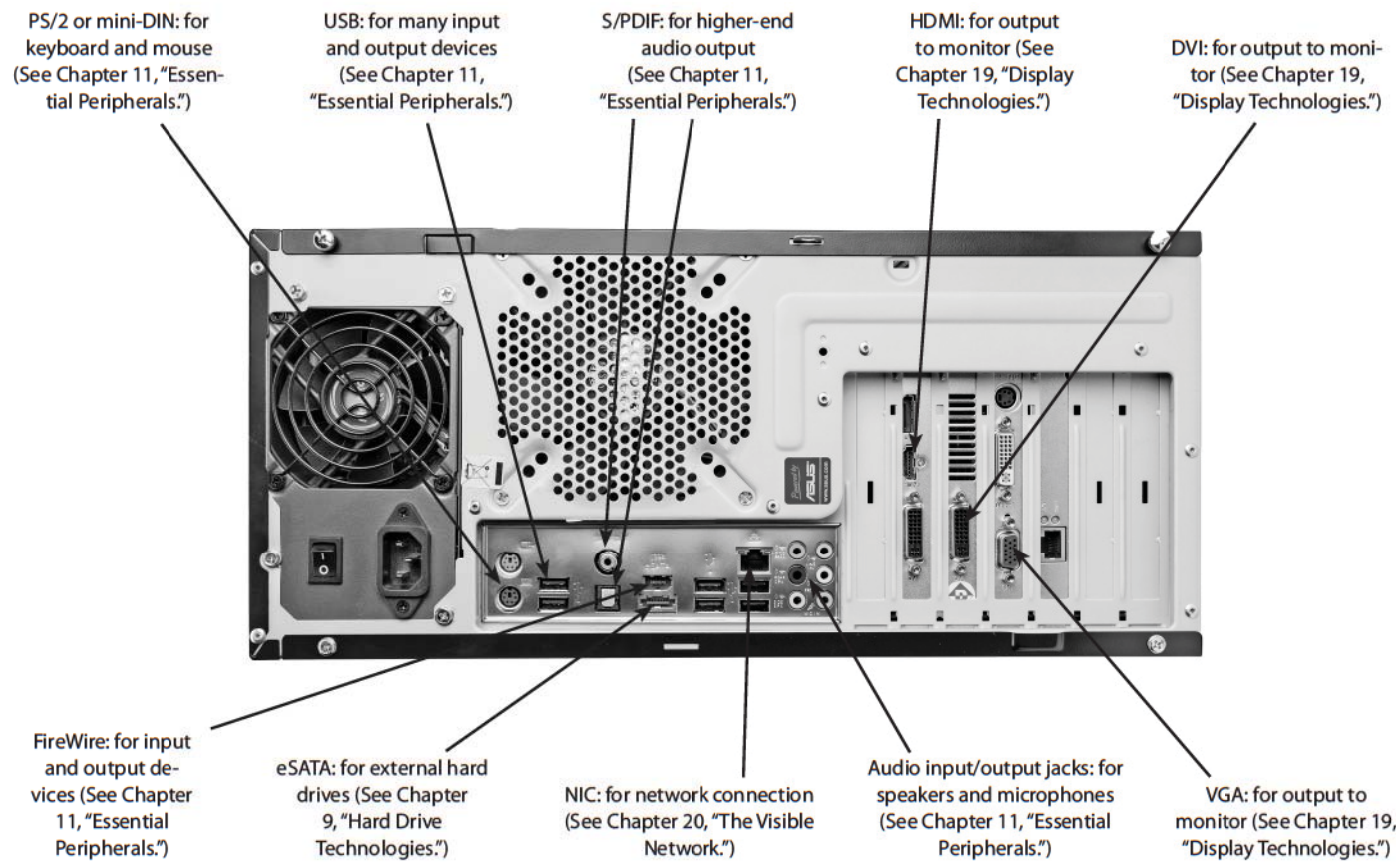


Figure 3-13 The business end of a PC

Figure 3-14 reveals the inside of a PC case, where you'll find the processing and storage devices. Hiding under everything is the motherboard, the component into which everything directly or indirectly connects.

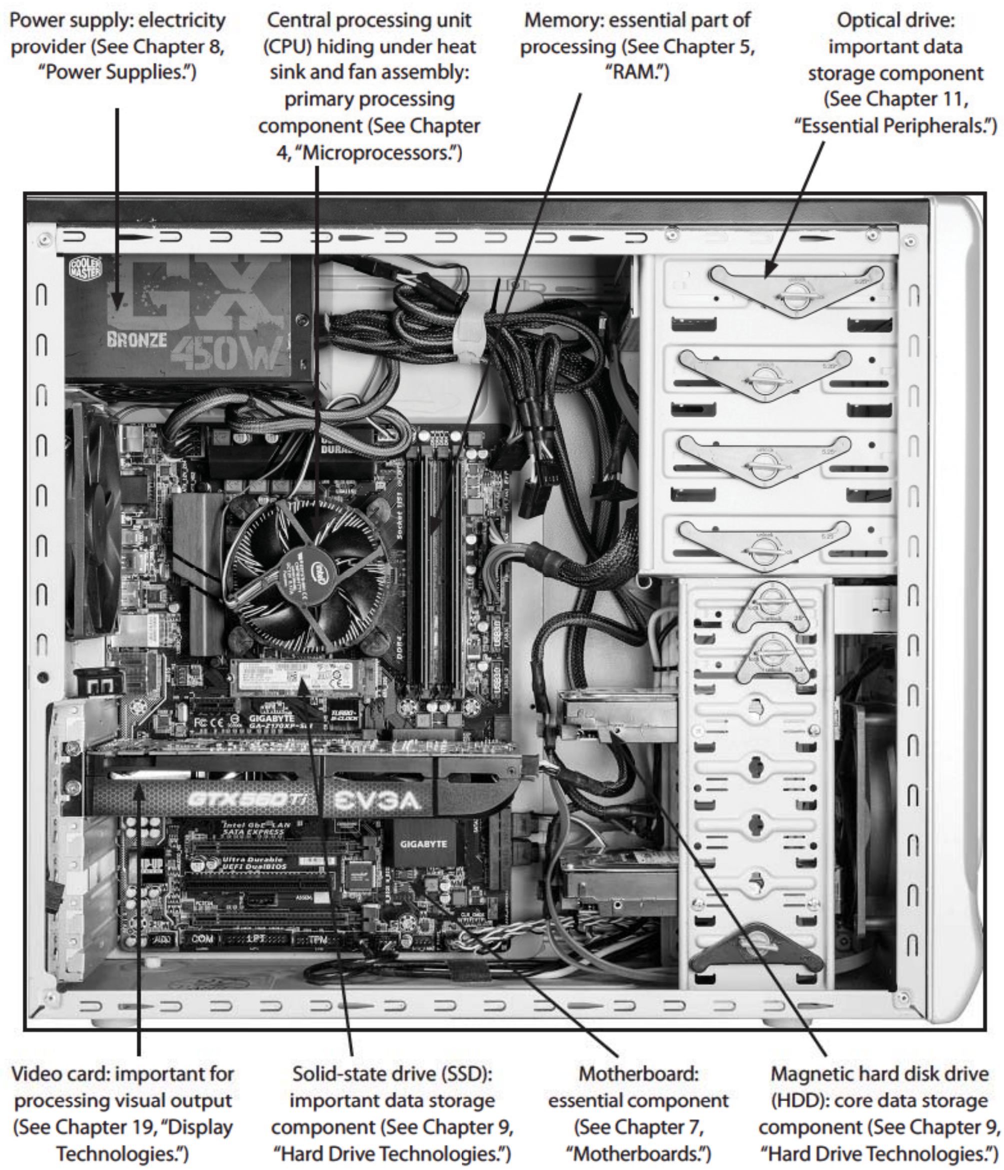


Figure 3-14 Inside the system unit

Figure 3-15 shows a clamshell-style portable computer, in this case an Apple MacBook Air. The portable nature of the device calls for input and output devices built into the case—some variation from the typical PC displayed earlier, therefore, but all the standard computing component functions apply. Chapter 24, "Portable Computing," goes into a lot of detail about each component displayed here.

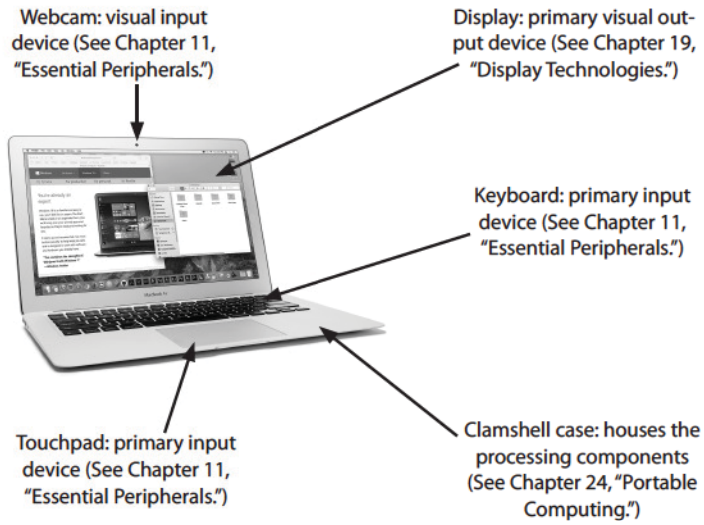


Figure 3-15 Portable computer (a MacBook Air)

Figure 3-16 shows the side of a portable computer with three different connection types.

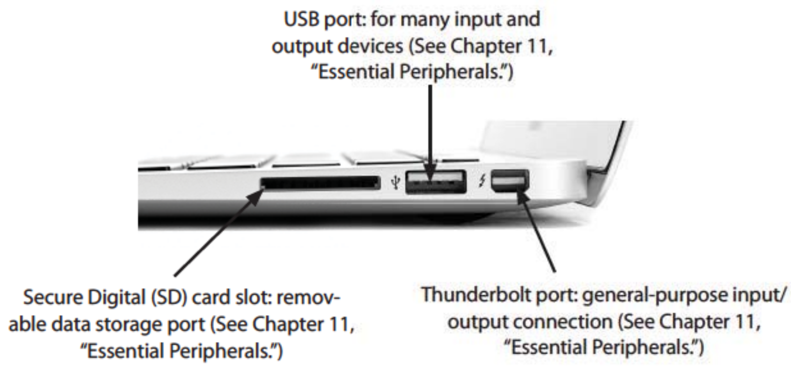
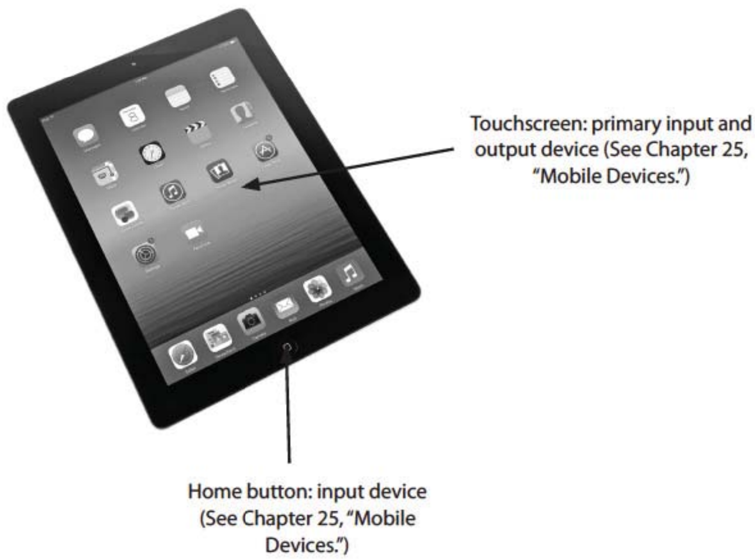


Figure 3-16 Ports on a portable computer

Figure 3-17 shows a tablet computer, an Apple iPad. Note that the screen has a touch interface, which makes it both an input and output device.

Figure 3-17
Tablet computer



We could continue with any number of computing devices in the same picture show, but at this point the uniformity of computing component functions should be pretty clear. They all work similarly, and, as a competent tech, you should be able to support just about any customer device. Let's turn now to a visual feast of software.

SIM Check out the excellent Chapter 3 Challenge! sim on motherboard matching at totalsem.com/90x. It's a cool sim that helps names stick in your head.

902

Computing Software

The CompTIA A+ 902 exam covers a lot of software, though mostly operating system tools rather than specific applications. Five Microsoft operating systems make up the bulk of the coverage: Windows Vista, Windows 7, Windows 8, Windows 8.1, and Windows Phone/Mobile. Windows 10 didn't make it into the objectives, but I've added it to the book because you need to know it for real-life support. Apple gets coverage of two OSs: OS X and iOS. Linux gets a generic nod (more on that in a moment), and Google Android gets some discussion.

NOTE Along with the unfortunately missing Windows 10, the CompTIA A+ objectives have a few notably absent operating systems. Microsoft released Windows 10 well after CompTIA finalized the exam objectives, so it gets no coverage. Perhaps more puzzling, though, is the lack of specific Linux versions, called *distributions* or *distros*. There's a big difference in look and feel, for example, between Ubuntu and SUSE Linux. Focus here on what you can do with every OS and you'll be able to handle any distro easily. Finally, Google Chrome OS, used on Google's line of portable computers (Chromebooks), gets nary a nod.

Common Operating System Functions

All OSs are not created equal, but every OS provides certain functions. Here's a list:

- The OS communicates, or provides a method for other programs to communicate, with the hardware of the PC or device. Operating systems run on specific hardware. For example, if you have a 32-bit computer, you need to install a 32-bit version of an operating system. With a 64-bit computer, you need a 64-bit OS.
- The OS creates a *user interface (UI)*—a visual representation of the computer on the monitor that makes sense to the people using the computer.
- The OS enables users to determine the available installed programs and run, use, and shut down the programs of their choice.
- The OS enables users to add, move, and delete the installed programs and data.
- The OS provides a method to secure a system from all sorts of threats, such as data loss or improper access.

Almost every chapter in this book explores the interaction of OS and hardware. Chapter 12, "Building a PC," examines adding and removing programs. Many security features show up in multiple chapters, such as Chapter 14, "User and Groups," and Chapter 27, "Securing Computers." The rest of this chapter, therefore, focuses on the user interface and the file structures.

User Interfaces

This section tours the various operating system user interfaces. Like the hardware tours earlier, this section serves a double purpose. First, you need to know the proper names for the various UI features and have an understanding of their functions. Second, it serves as a handy quick review section before you take the 902 exam.

EXAM TIP Be sure you are very familiar with the operating system feature names, tools, and terms discussed and displayed in this section. Not only will you see them in future chapters, you will also encounter them in the field as well as in the CompTIA A+ 902 exam.

NOTE Chapter 25, "Mobile Devices," details the three operating systems for mobile devices—iOS, Android, and Windows Phone/Mobile.

Windows Vista/7

Figure 3-18 shows the standard interface for Windows 7, a traditional multifunction computer. Windows uses a graphical user interface primarily, so you engage with the mouse or other pointing device and click on elements. The background is called the *Desktop*. The open applications are Internet Explorer—Window's default Web browser—and a Windows Explorer window showing the Windows 7 default Libraries.

Other visible items are as follows:

- The open applications demonstrate *transparency*, where the edges of the applications show blurred background images. This feature is called *Aero*, or *Aero Glass*.
- Click on the *Start button* to get access to applications, tools, files, and folders.
- The *pinned programs* enable you to launch a program with a single left-click.
- The *taskbar* shows running programs.
- The *notification area* shows programs running in the background. Many techs also calls it the *system tray*.

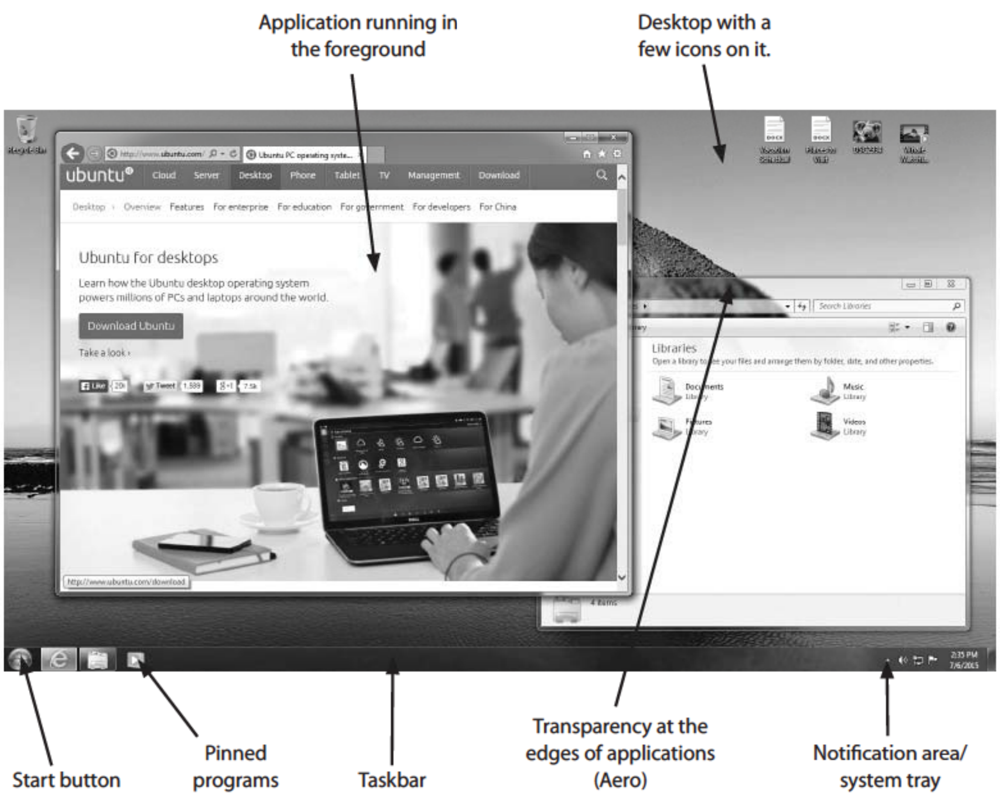


Figure 3-18 Windows 7 with applications open

Interacting with the classic Windows interface for the most part involves using a mouse or touchpad to move the cursor and either left-clicking or right-clicking on the icons. Left-clicking selects an item; double-left-clicking opens an item. Right-clicking opens a *context menu* from which you can select various options (see Figure 3-19).

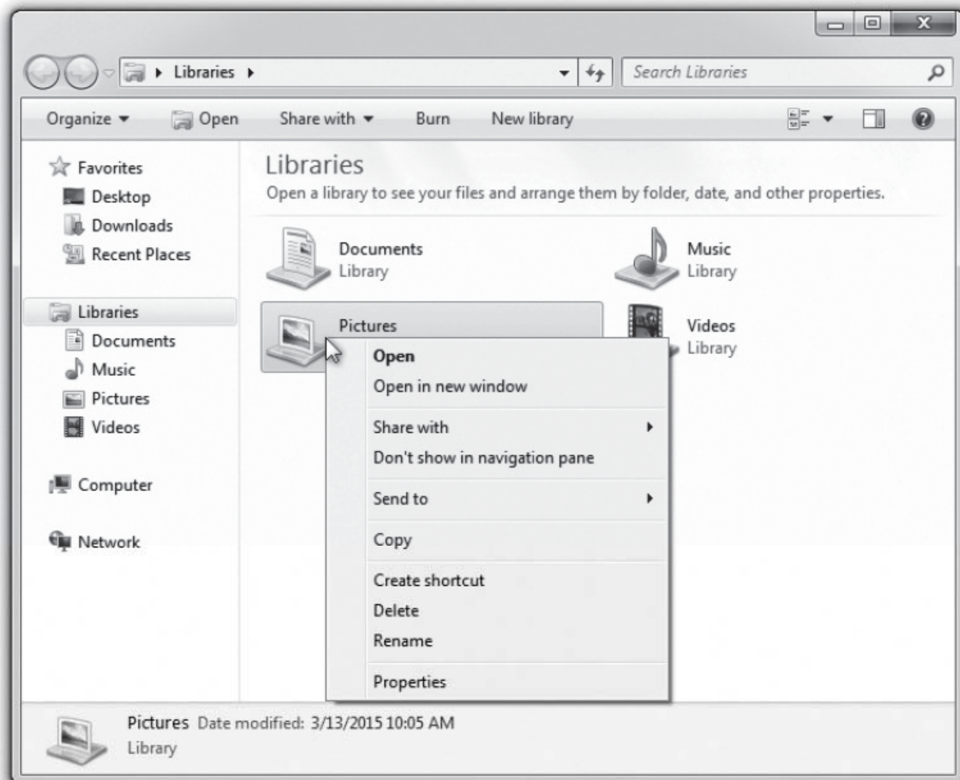


Figure 3-19 Context menu

NOTE The context menu offers options specific to the icon you right-click. Right-clicking on a file, for example, gives you a context menu that differs greatly from when you right-click an application.

Windows 7's predecessor, Windows Vista, has a similar look and feel. The most visible difference is the Vista feature called the Sidebar. Enabled by default, the *Sidebar* houses one or more *Gadgets*, such as the Clock, Calendar, and speeds you can see in Figure 3-20. Windows 7 supports Gadgets too, but doesn't have a Sidebar.

CAUTION Because of the inherent security flaws with both Sidebars and Gadgets, Microsoft recommends disabling them on Windows Vista and Windows 7 systems. Later versions of the OS do not have either.

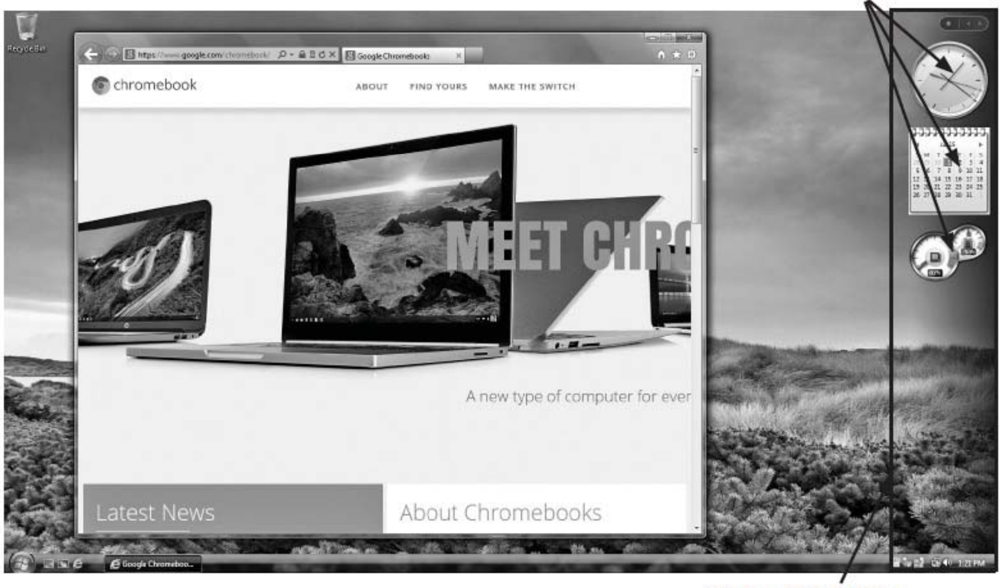


Figure 3-20 Windows Vista

Windows 8/8.1

Microsoft made significant changes to the Windows interface with the introduction of Windows 8. They borrowed from tablet operating systems, such as Windows Phone, to create a graphical set of *tiles* for full-screen programs, called *apps*. Note that the screen shows *pinned apps*—the default programs and programs selected by the user—and not all the applications installed on the computer.

The Windows 8 interface, code-named *Metro UI*, works great for touch-enabled devices. The PC becomes in essence a giant tablet. Touch an app to load, drag your finger across the screen to see other apps, and have fun. Figure 3-21 shows the default Windows 8 interface, called the *Start screen*, with various elements called out.

NOTE Microsoft dropped the "Metro UI" moniker just before releasing Windows 8 due to legal concerns, replacing it with "Modern UI." A lot of techs and IT industry pros continue to refer to the unique Windows 8/8.1 tiled interface as "Metro."

Windows 8 also features a more classic Desktop, but one with the noticeable absence of a visible Start button (see Figure 3-22). You access this screen by pressing the *Windows logo key* on a standard keyboard.

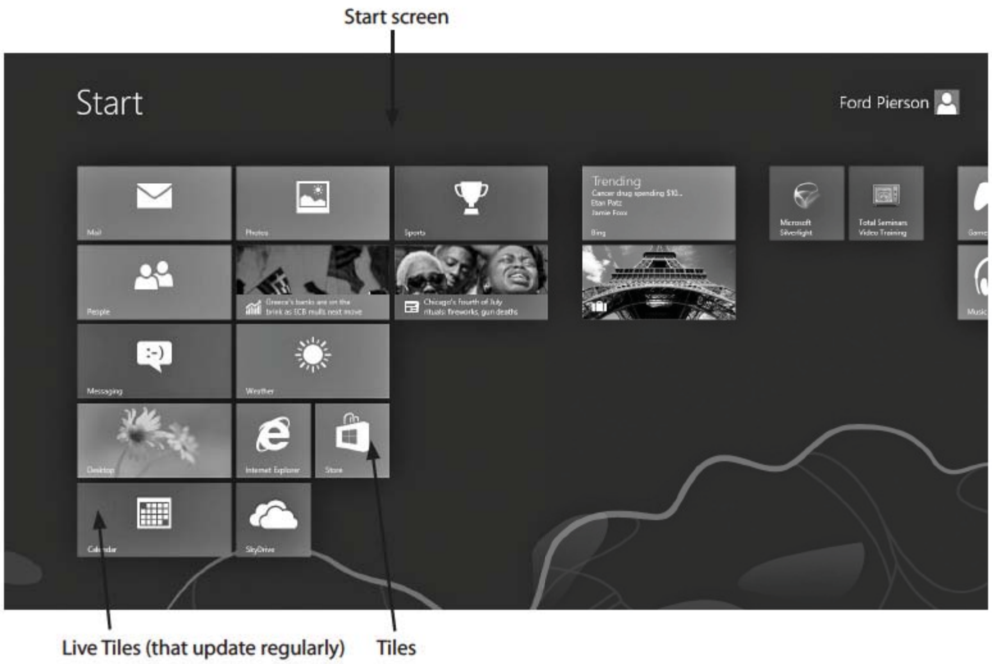


Figure 3-21 Windows 8 Start screen

Figure 3-22 Windows 8 Desktop

Figure 3-23 Windows 8 Start screen scrolled to the right

Using a keyboard and mouse with Windows 8 bothers a lot of users making the jump from Windows 7. Scrolling with the mouse wheel, for example, scrolls right to left rather than up and down (see Figure 3-23).

With a series of updates culminating in Windows 8.1, Microsoft brought back features such as the Start button, easy access to a Close button for apps, and the ability to boot directly to the Desktop. Figure 3-24 shows the standard interface for Windows 8.1 with the various elements called out. Note that it's very similar to Windows 7.

Windows 8.1 makes it very easy to pin apps to the Start screen. Selecting the arrow at the bottom left brings up the Apps pane where you can sort and select apps and utilities (see Figure 3-25). Right-click on an icon to pin it to the Start screen.

Windows 8/8.1 offer lots of hidden interface components that activate when you place the cursor in certain places on the screen. Dropping the cursor to the bottom left corner, for example, activates the Start button (see Figure 3-26) when in the Start screen.

EXAM TIP The first release of Windows 8 had no visible Start button on the Desktop (except in the Charms bar). Microsoft added it to the Desktop in later patches.

Placing the cursor in the top- or bottom-right corner of the screen reveals the *Charms bar*, a location for tools called *charms*. See the right side of Figure 3-27. Charms include a robust Search tool that enables a search of the computer or even the Internet in one location. There's a Share charm for sharing photos, e-mail messages, and more. We'll revisit the charms later in this chapter when exploring how to access tech tools.

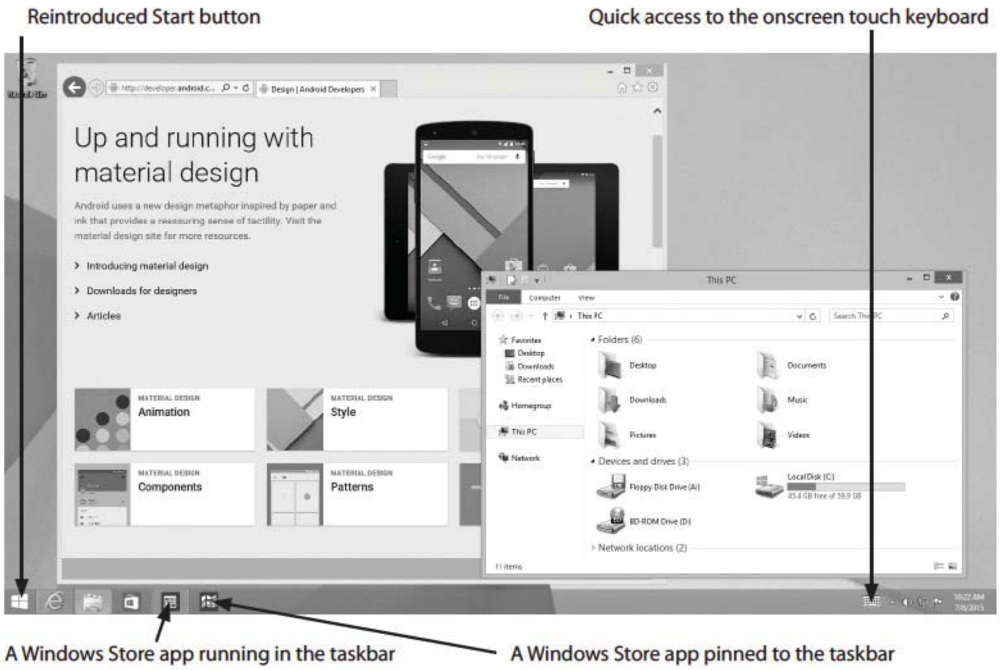


Figure 3-24 Windows 8.1

Figure 3-25 Apps sorted by name

Figure 3-26 Start button magically appears

Figure 3-27 Charms accessed by cursor in upper- or lower-right corner

Figure 3-28 Windows 8.1 Desktop

The final version of Windows 8.1 uses the Desktop rather than the Start screen as the default interface. The Start button is visible in the bottom left (see Figure 3-28). You can still access the charms using the cursor and the upper- and lower-right corners of the screen.

Windows 10

With Windows 10, Microsoft created an OS that blends the traditional Windows 7–style Desktop experience with some of the more progressive features of the Windows 8.*x* Metro/Modern UI. In particular, Microsoft brought the Start menu back with conviction. They removed the much unloved Charms bar. Microsoft incorporated the essential tools—Search being my go-to feature—into the desktop in the lower-left corner of the taskbar. Figure 3-29 shows the Windows 10 interface with an active application in the foreground.

When you press the WINDOWS KEY on the keyboard, Windows 10 brings up the Start menu with useful tools and your most used apps on the left and pinned apps on the right (see Figure 3-30). Just like with Windows 8.1, you can click on the link helpfully named All apps (bottom left) to open a list of installed applications. Right-click to pin any app to the Start screen. Windows 10 altered the side-by-side app feature introduced in Windows 8. Use the Windows key and right or left arrow key to flip an app to one side of the screen. Do the same on another. Sweet!

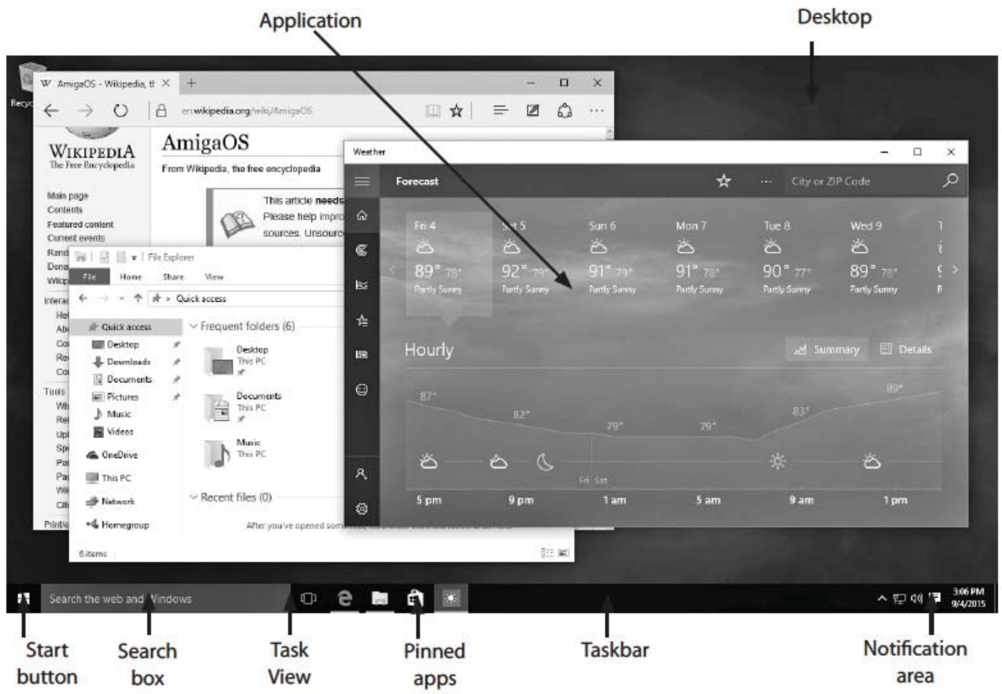


Figure 3-29 Windows 10 with a few applications open

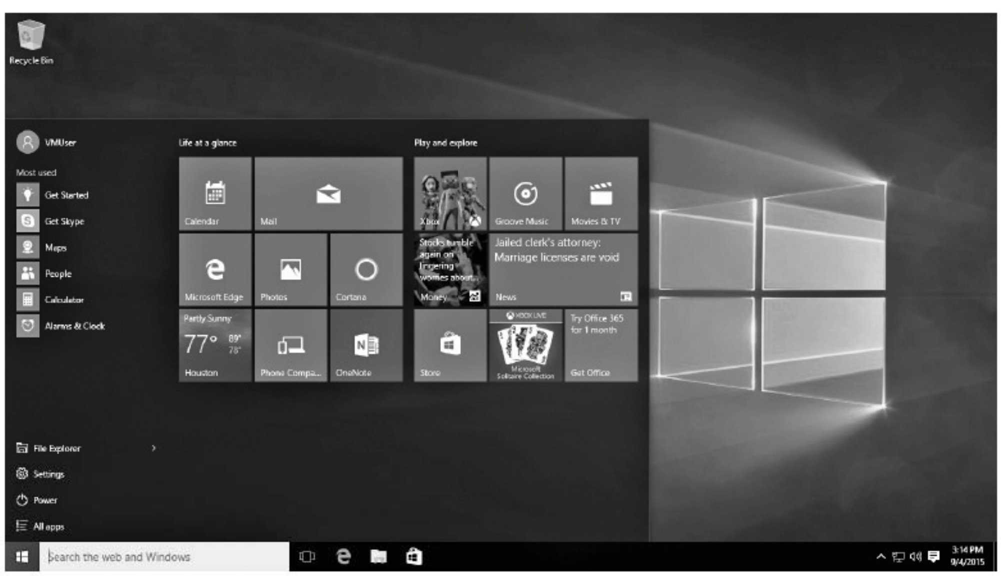


Figure 3-30 Start menu in Windows 10

Click on the Windows 10 Task View button to create and manage *multiple Desktops* for grouping your open applications. Mac OS X and Linux each have their own take on this feature, as you'll see in the following sections.

Mac OS X

The Mac OS X operating system interface offers similar functions to those found on Windows. The background of the main screen is called the *Desktop*. You can access frequently used applications by clicking on their icons on the *Dock*. Just like with the taskbar pinned apps, you can add and remove apps from the Dock with a right-click. The Dock is more than a set of apps, though. It also shows running applications (like the taskbar in Windows). Figure 3-31 shows a typical Mac OS X interface.

Pressing the Mission Control button on an Apple keyboard (see Figure 3-32) brings up a utility, called *Mission Control*, that enables you to switch between open applications, windows, and more, as shown in Figure 3-33. You can also access Mission Control by pressing and holding the control/ctrl key and pressing the up arrow key.

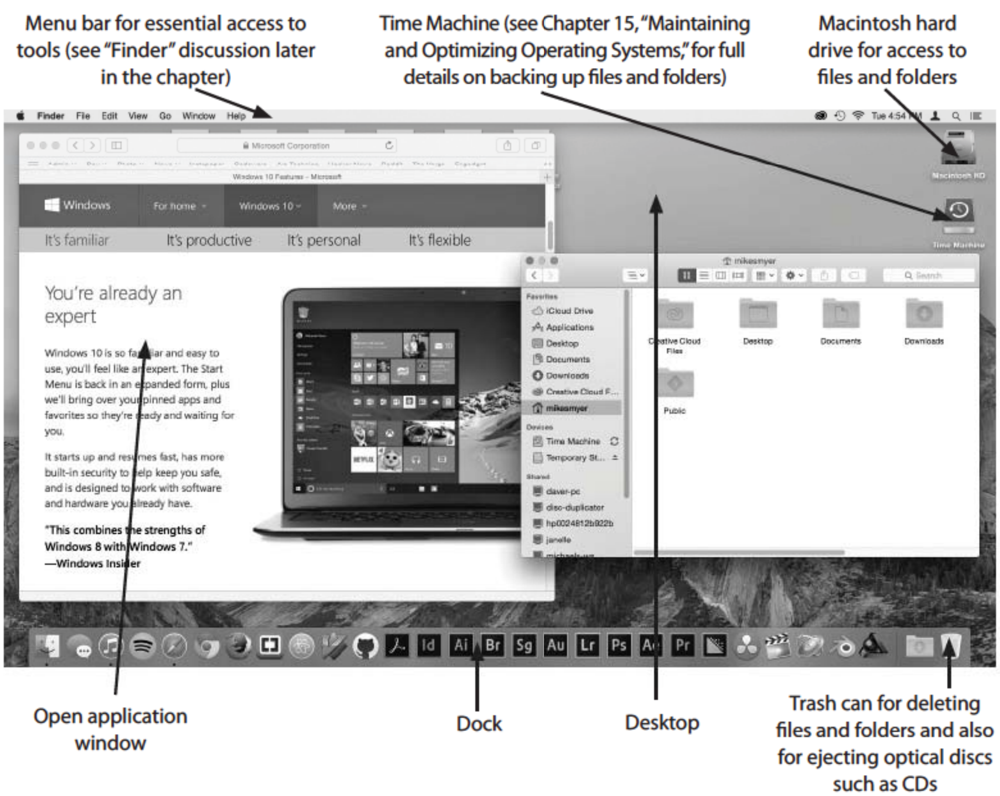


Figure 3-31 Mac OS X

Figure 3-32 Mission Control button on keyboard

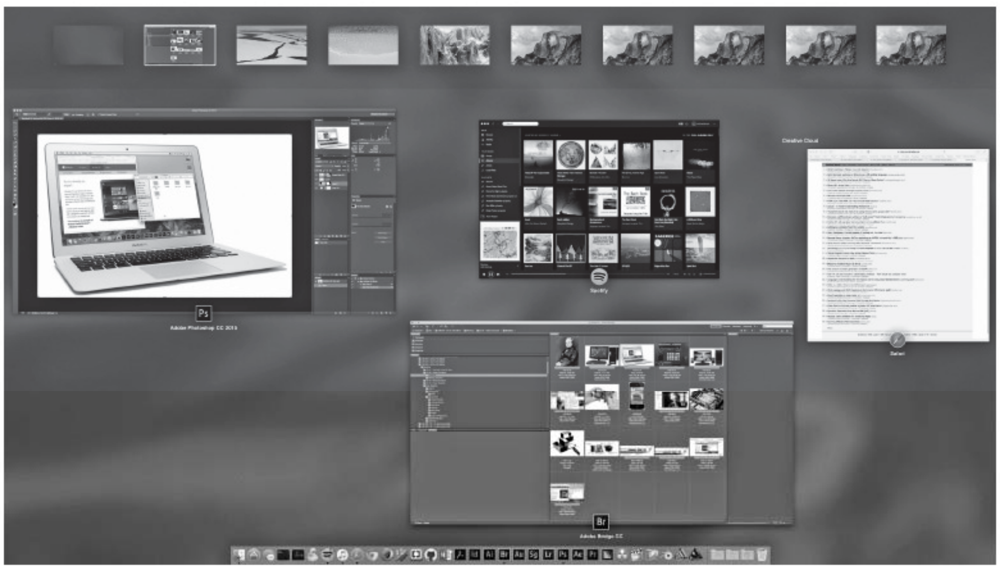


Figure 3-33 Mission Control showing four open apps and nine Desktops

Mac OS X supports *Spaces*—essentially multiple Desktops—that can have different backgrounds and programs (but keep the same Dock). You can optimize your workflow, for example, by putting your primary program full screen on Desktop 1 and putting your e-mail client on Desktop 2 (see Figure 3-34). New messages won't disturb you when working, but you can access the second Desktop easily when you want with Mission Control. On the latest versions of Mac OS X, press and hold the control key and press the right arrow and left arrow keys to scroll through Spaces.

EXAM TIP Windows 10 supports multiple Desktops with Task View, but you won't find support for that feature in earlier versions of Windows.

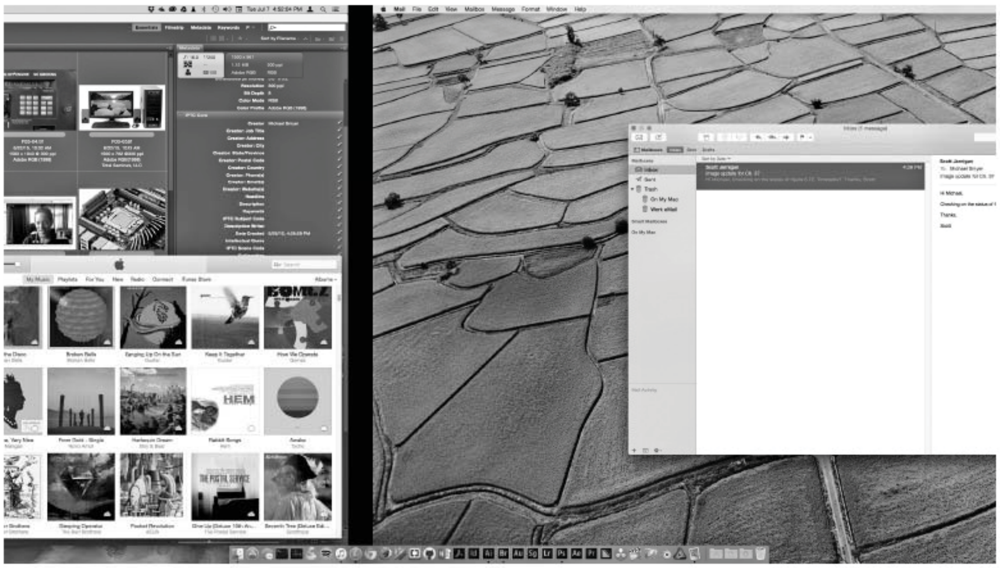


Figure 3-34 Switching between multiple Desktops

Linux

The many different distributions of Linux offer a variety of user interfaces, called *desktop environments (DEs)*, but they offer similar functions to those in Windows or Mac OS X. Figure 3-35 shows a popular Linux distro, Ubuntu Linux with the Unity DE, and notes the various features. Frequently used utilities and applications are locked on the Launcher on the left side of the screen. The top-left icon—the Ubuntu button—offers powerful system/network/Internet searching, while the next icon down enables you to access files and folders.

> **Try This!**
>
> **Ubuntu Emulator Online**
>
> Ubuntu.com has a fairly robust emulator for Ubuntu Linux that enables you to poke around the desktop, check out settings and so forth. Try This! Open www .ubuntu.com, type **tour** in the Search option on the page, and press ENTER. In the search results, click on the first link to Take the tour. Have fun!

File Structures and Paths

Knowing where to find specific content—files and the folders in which they reside—helps techs help users do their day-to-day tasks more efficiently. Almost every operating system stores files in folders in a tree pattern. The root of the tree is the drive or disc,

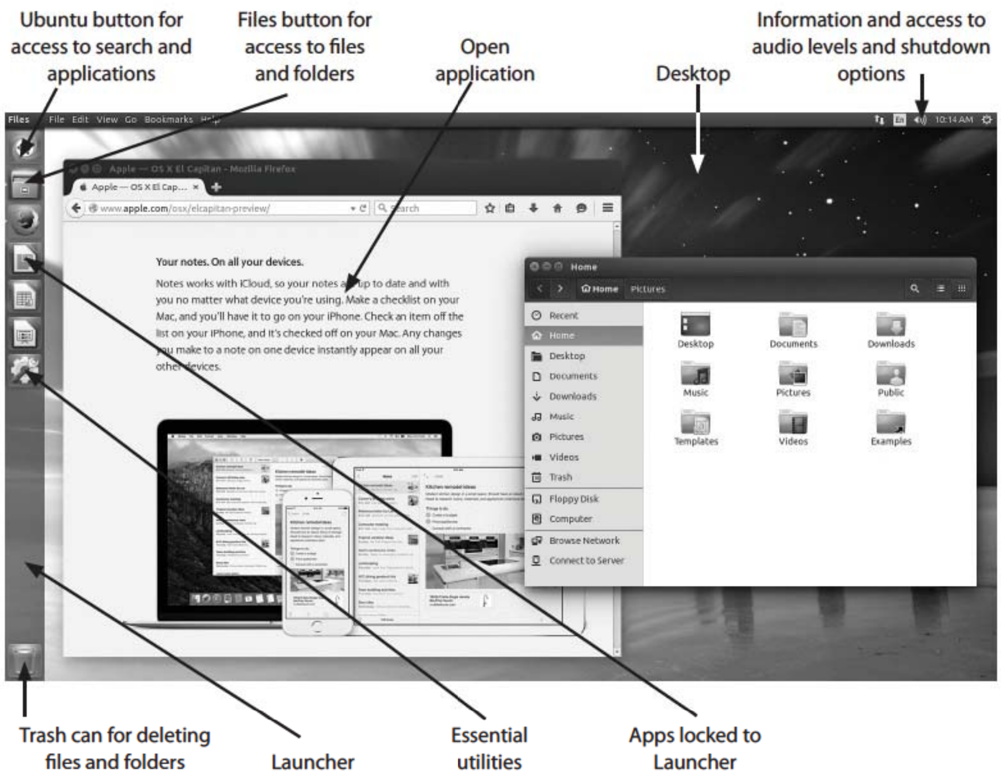


Figure 3-35 Ubuntu Linux

followed by a folder, subfolder, sub-subfolder, and so on, until you get to the desired file. The drive or disc gets some designation, most usually a *drive letter* like C:. Chapter 10, "Implementing Hard Drives," goes into gory detail on how modern operating systems implement systems for storing data. This section is more dictated by CompTIA's obsession with requiring examinees to memorize paths.

Windows

Windows has a number of important folders that help organize your programs and documents. They sit in the *root directory*—where the operating system is installed—and of course they have variations depending on the version of Windows. The following sections walk through the locations of important folders.

Most users and techs access folders and files in Windows with a tool called *Windows Explorer* in Windows Vista/7 and *File Explorer* in Windows 8/8.1/10—although you can only see that difference in name by right-clicking on the Start button or by moving your mouse over the folder icon in the taskbar (see Figure 3-36).

Figure 3-37 shows File Explorer viewing the Desktop in Windows 8. Select View to change Folder Options, such as view hidden files, hide file extensions, general options, and other view options.

Figure 3-36 Mousing over the File Explorer icon

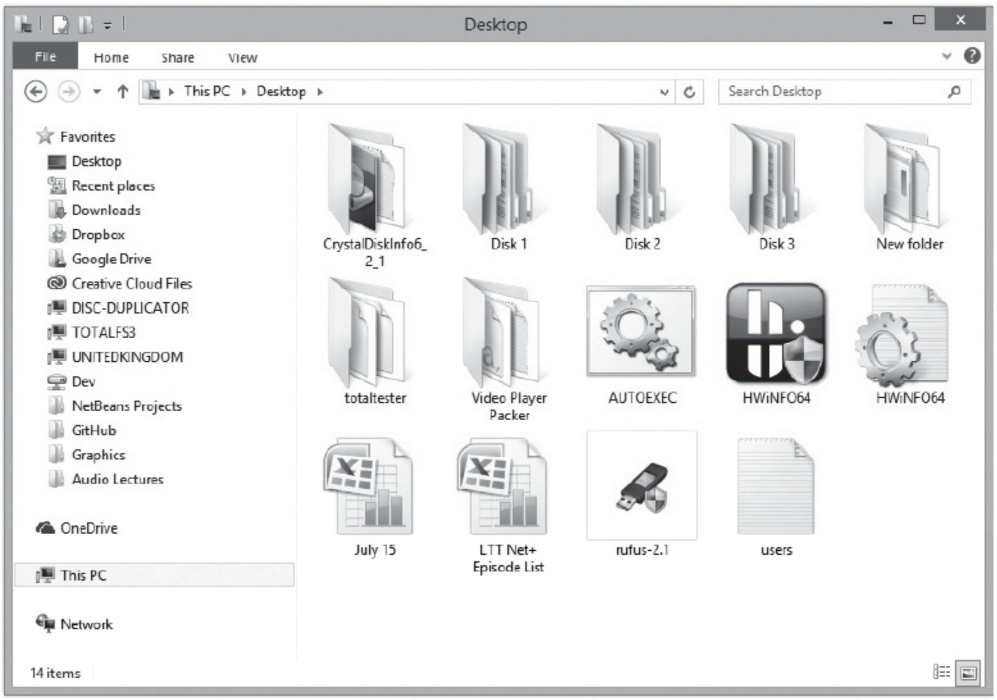


Figure 3-37 File Explorer

The folder structures that follow here use the standard formatting for describing folder structures. This is what you'll see on the 902 exam and in almost any OS. Windows hides the "\" characters at the beginning to make it prettier. File Explorer might show something like "Local Disk (C:) > Users > Mike." This translates in proper fashion as C:\Users\Mike.

C:\Program Files (All Versions) By default, most programs install some or all of their essential files into a subfolder of the Program Files folder. If you installed a program, it should have its own folder in here. Individual companies decide how to label their subfolders. Installing Photoshop made by Adobe, for example, creates the Adobe subfolder and then an Adobe Photoshop subfolder within it.

C:\Program Files (x86) The 64-bit editions of Windows create two directory structures for program files. The 64-bit applications go into the C:\Program Files folder, whereas the 32-bit applications go into the C:\Program Files (x86) folder. The separation makes it easy to find the proper version of whatever application you seek.

Personal Documents Modern versions of Windows use subfolders of the C:\Users folder to organize files for each user on a PC. Figure 3-38 shows the default folders for a user named Mike. Let's quickly survey the ones you need to know for the CompTIA A+ exams:

- **C:\Users\Mike\Desktop** This folder stores the files on the user's Desktop. If you delete this folder, you delete all the files placed on the Desktop.
- **C:\Users\Mike\Documents** This is the Documents or My Documents folder for that user. Only Windows 7 uses My Documents. The others use Documents.
- **C:\Users\Mike\Downloads** Microsoft's preferred download folder for applications to use. Most applications use this folder, but some do not.
- **C:\Users\Mike\Music** This is the default location for music you download. My guess is that more people have music in iTunes, but that's just me.

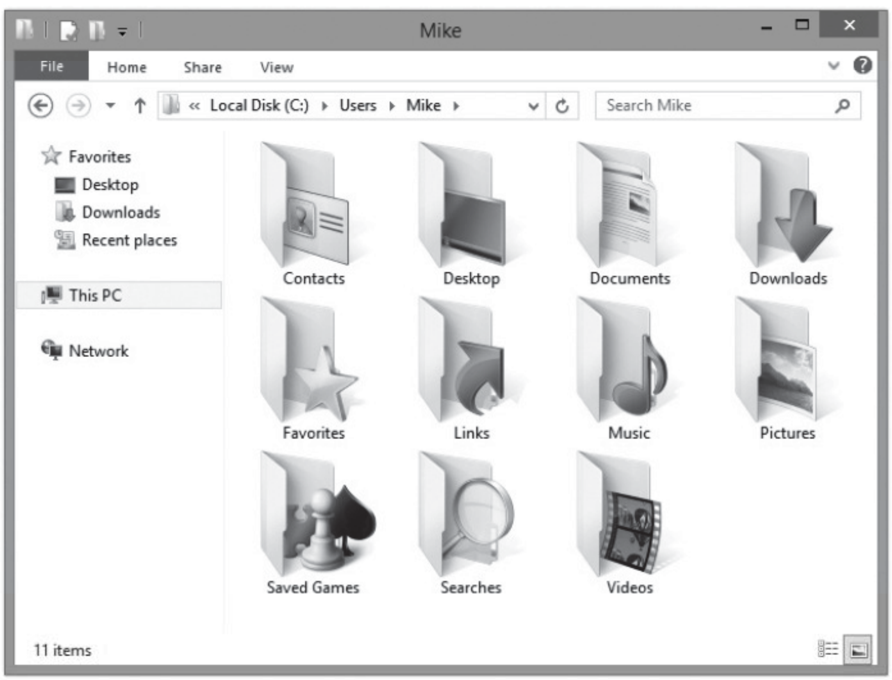


Figure 3-38 File Explorer viewing Mike's folders

- **C:\Users\Mike\Pictures** Pictures is the default location for images imported into the PC, although the Pictures library can (and does) draw from many folder locations.
- **C:\Users\Mike\Videos** Videos is the default location for movies and homebrewed videos imported into a PC.

Mac OS X

Finder holds the keys to files and folders in Mac OS X. Figure 3-39 shows Finder open to display Mike's Users folder. Note that although its style differs from the Windows screen earlier, it has functionally similar folders. These are the default locations for files on the Desktop, in Documents, Downloads, Music, Pictures, and so on. Each user account on the Mac will have a unique Users folder that is inaccessible by other users on that computer.

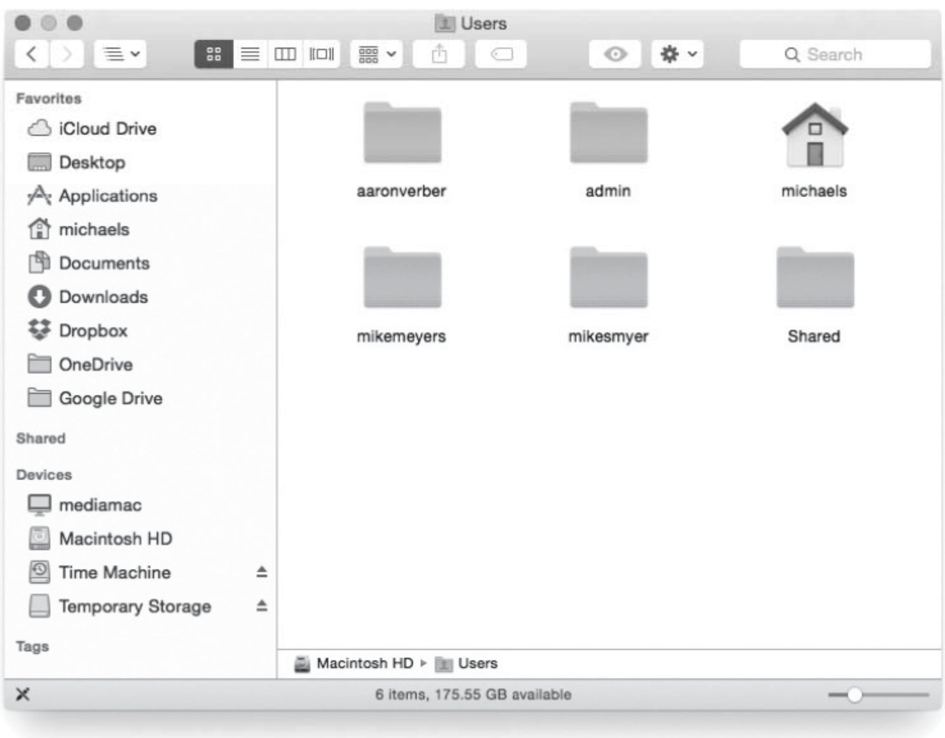


Figure 3-39 Finder

Linux

Ready to be shocked? Not surprisingly, Linux uses pretty much the same structure for user organization (see Figure 3-40). I guess once something seems logical to enough people, there's no reason to add confusion by changing the structure. The only major difference is the name: Linux uses the Home folder, rather than the Users folder.

The Tech Launch Points

Every OS has two or three areas for tech-specific utilities. This section shows you how to access those areas, primarily so that we don't have to repeat the steps to get to them when accessing them many times throughout the book. Just refer back to this section if you have difficulty remembering how to arrive at a place later on. Also, CompTIA will test your knowledge on how to access these tool locations, with specific steps. Use this section for the last-minute cram before taking the exams.

EXAM TIP The 902 exam will test you on specific paths to specific tools. You will get several of these questions as multiple-choice, scenario-based, or both types of question.

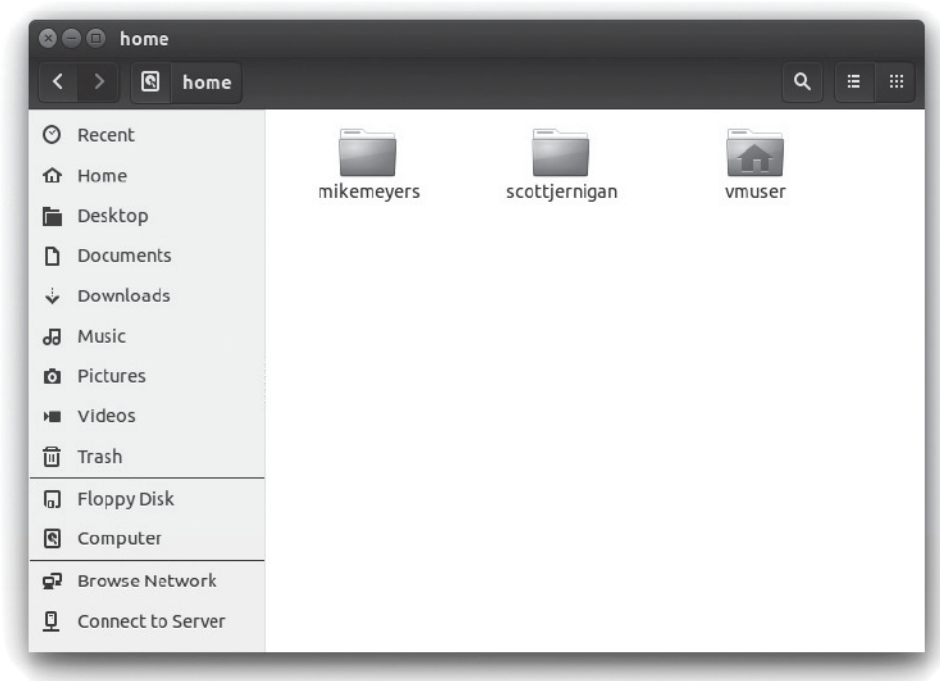


Figure 3-40 Home folder

Windows Vista/7

Windows Vista/7 have three tech launch points: the Control Panel, System Tools, and the command-line interface. You can get to each launch point in multiple ways.

Control Panel The *Control Panel* handles most of the maintenance, upgrade, and configuration aspects of Windows. As such, the Control Panel is the first set of tools for every tech to explore. You can find the Control Panel by clicking on the Start button and choosing Control Panel from the Start menu.

The Control Panel opens in the Control Panel's Category view by default, which displays the icons in groups like Hardware and Sound. See Figure 3-41. This view requires an additional click (and sometimes a guess about which category includes the applet you need), so many techs use Classic view.

The CompTIA A+ 902 exam specifically assumes Classic view with large icons, so you should do what every tech does: switch from Category view to Classic view. In Windows Vista, choose Classic View. In Windows 7, select either Large icons or Small icons from the View by drop-down list for a similar effect. Figure 3-42 shows the Windows Vista Control Panel in Classic view.

A large number of programs, called *applets*, populate the Control Panel. The names and selection of applets vary depending on the version of Windows and whether any installed programs have added applets. But all versions of Windows have applets that enable you to control specific aspects of Windows, such as the appearance, installed

Figure 4-41 Windows 7 Control Panel (Category view)

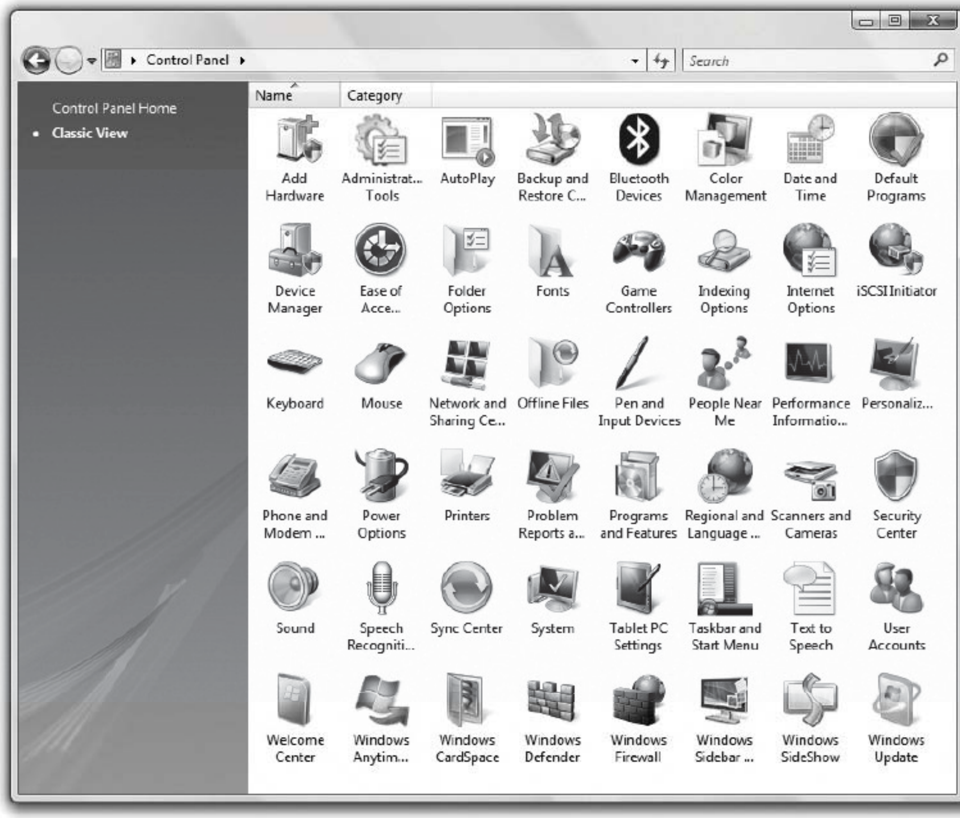


Figure 3-42 Windows Vista Control Panel (Classic view)

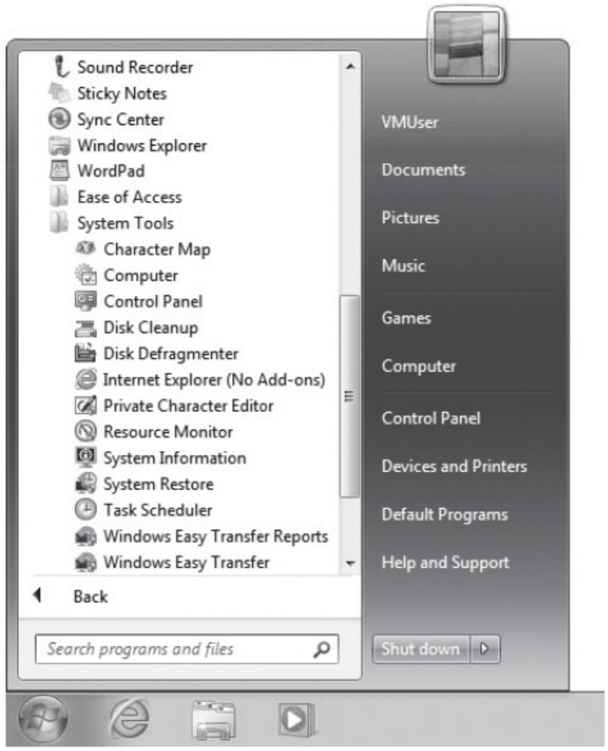
applications, and system settings. You will get details on each applet as we put them into use over the course of this book.

System Tools The Start menu offers a variety of tech utilities collected in one place: System Tools. In the *System Tools* menu, you'll find commonly accessed tools such as System Information and Disk Defragmenter (see Figure 3-43).

Many techs overlook memorizing how to find the appropriate Windows tool to diagnose problems, but nothing hurts your credibility with a client like fumbling around, clicking a variety of menus and applets, while mumbling, "I know it's around here somewhere." The CompTIA A+ certification 902 exam therefore tests you on a variety of paths to appropriate tools.

To access System Tools in Windows Vista/7, go to Start | All Programs | Accessories | System Tools. Each version of Windows shares many of the same tools, but each includes its own utilities as well. Rather than go through every tool here, I'll discuss each in

Figure 3-43 System Tools menu options



detail during the appropriate scenarios in the book. Here's one example that won't appear again, Character Map.

Ever been using a program only to discover you need to enter a strange character such as the euro character (€) but your word processor doesn't support it? That's when you need the Character Map. It enables you to copy any Unicode character into the Clipboard (see Figure 3-44) and paste into your document. Unicode has all the special symbols and alphabet characters used in languages throughout the world.

Command Line The Windows *command-line interface* is a throwback to how Microsoft operating systems worked a long, long time ago when text commands were entered at a command prompt. Figure 3-45 shows the command prompt from DOS, the first operating system commonly used in PCs.

DOS is dead, but the command-line interface is alive and well in every version of Windows. Every good tech knows how to access and use the command-line interface. It is a lifesaver when the graphical part of Windows doesn't work, and it is often faster than using a mouse if you're skilled at using it. An entire chapter (Chapter 16, "Working with the Command-Line Interface") is devoted to the command line, but let's look at one example of what the command line can do. First, you need to get there. Click on

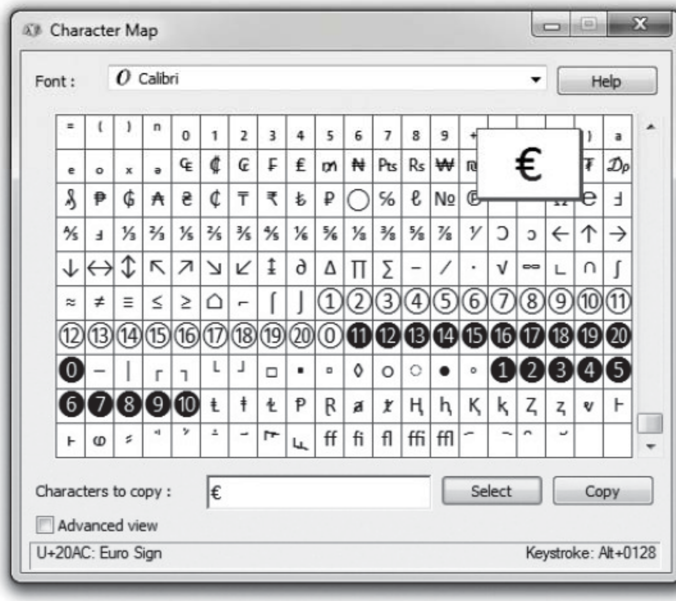


Figure 3-44 Character Map

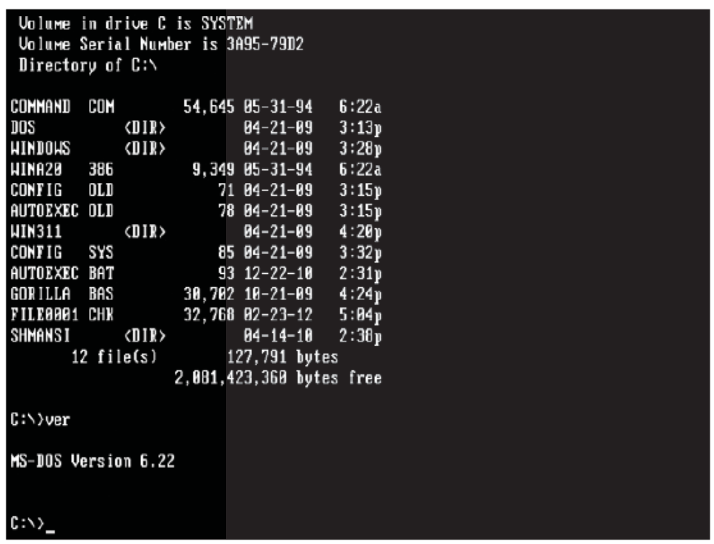


Figure 3-45 DOS command prompt

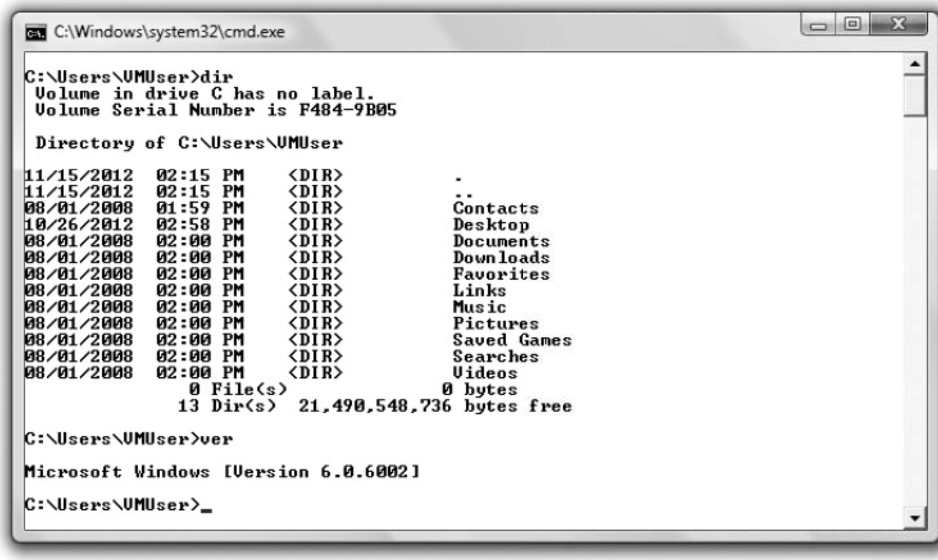


Figure 3-46 Command prompt in Windows Vista

the Start button, type **cmd** in the Search text box, and press the ENTER key. Figure 3-46 shows a command prompt in Windows Vista.

Once at a command prompt, type **dir** and press ENTER. This command displays all the files and folders in a specific directory—probably your user folder for this exercise—and gives dates, times, folder names, and other information. The dir command is just one of many useful command-line tools you'll learn about in this book.

Windows 8/8.1

Windows 8/8.1 have three tech tool starting points, but they differ a little from the big three in Windows Vista/7. The newer versions feature the Control Panel, Administrative Tools, and the command-line interface.

Control Panel The Control Panel in Windows 8/8.1 serves the same function as in previous versions of Windows—the go-to source for tech tools. You can access the Control Panel in several ways:

- Tap the down arrow on the lower right of the Start screen and scroll all the way to the right in the list of Apps. In the Windows System category, click on Control Panel (see Figure 3-47). That's the slow way, but you should know it for the exams. You can also start typing **control panel** in the Search field in the Apps list. Control Panel will quickly appear as the best option to select.
- Right-click on the Start button and select Control Panel from the menu (see Figure 3-48). You can bring up the same menu by pressing WINDOWS KEY-X.

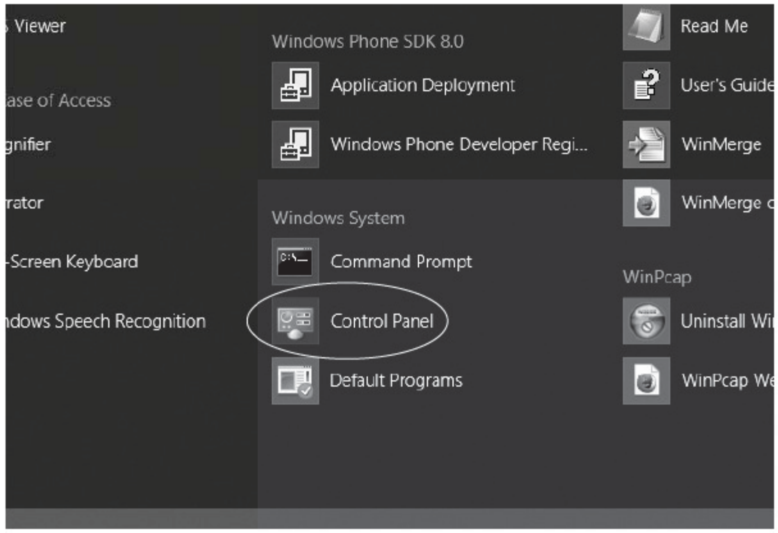


Figure 3-47 Selecting Control Panel from the list of Apps

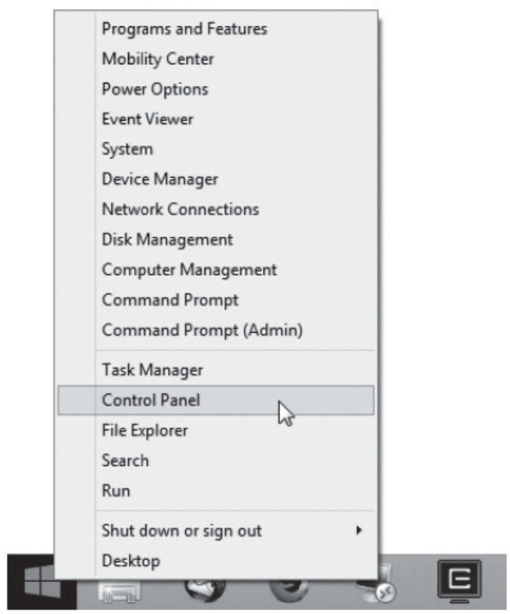
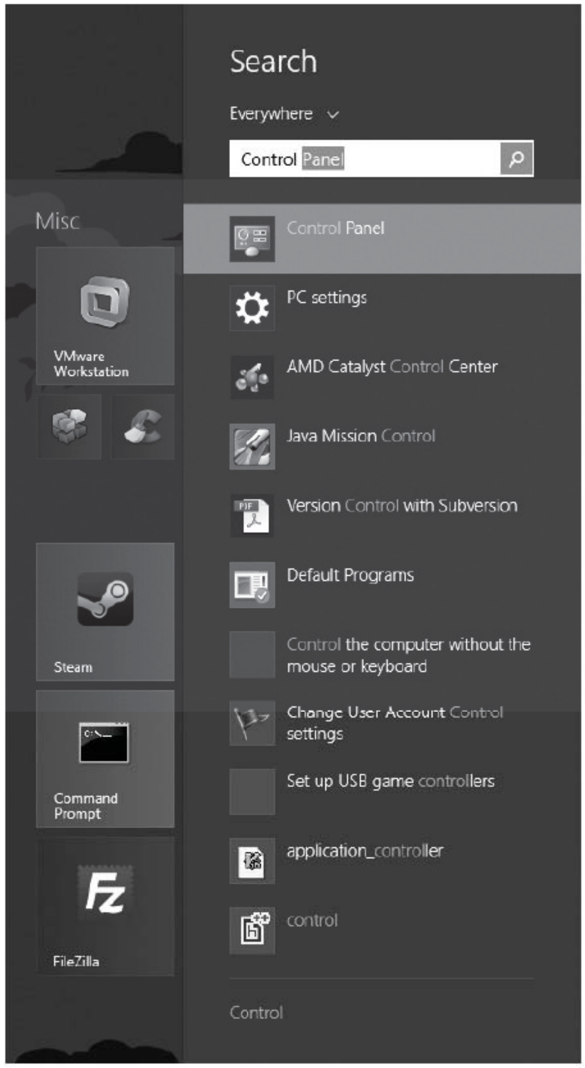


Figure 3-48 Right-clicking on the Start button

I call this menu *Tech Essentials* because it gives you very quick access not only to the Control Panel and its collection of tools but also to specific tools that every tech relies on heavily, like the Task Manager (for forcing frozen programs to close, among other things).

- In the Start screen, start typing **control panel**; the Control Panel will show up as the top option in the Search charm (see Figure 3-49). Select it to open.

Figure 3-49
Search charm with Control Panel as top option



Administrative Tools Microsoft beefed up Administrative Tools starting in Windows 8, adding some of the tools found in the System Tools menu in previous versions of Windows. *Administrative Tools* enables you to set up hard drives, manage devices, test system performance, and much more. This is a go-to set of tools for every tech, and one that we will access many times for scenarios in this book.

As with Control Panel, you have several options for accessing Administrative Tools:

- In the Start screen, click on the down arrow to open the Apps list. Scroll a little to the right and you'll see the list of Administrative Tools (see Figure 3-50). Select the specific tool you want to open.

Figure 3-50 Administrative Tools in the Apps list

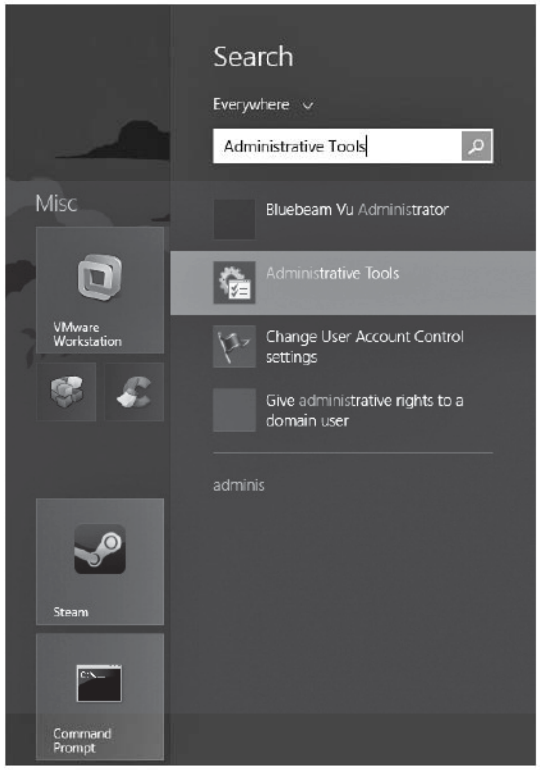
- Begin typing **administrative tools** in the Start screen and Administrative Tools will quickly appear as an option in the Search charm (see Figure 3-51). Select it to open.

- Right-click on the Start button (or press WINDOWS KEY-X) and select Control Panel from the context menu. In the Control Panel, select Administrative Tools to open.

Command Line The command-line interface retains its place as a go-to tool for techs. Windows 8/8.1 offers several ways to access it:

- Tap the down arrow on the lower right of the Start screen and scroll all the way to the right in the list of Apps. In the Windows System category, click on Command Prompt to open the utility. You can also start typing **command prompt** in the Search field in the Apps list. Command Prompt will quickly appear as the best option to select.
- Right-click on the Start button (or press WINDOWS KEY-X) and select Command Prompt from the context menu to open the command-line interface.
- In the Start screen, start typing **cmd** or **command prompt** and Command Prompt will appear in the Search charm. Click on it to open it.

Figure 3-51 Administrative Tools option in the Search charm



Windows 10

Windows 10 keeps the Control Panel and command-line interfaces we see in earlier Windows versions, but focuses on an expanded Settings app for day-to-day administration.

Control Panel Windows 10 offers two standard ways to get to the Control Panel. Right-click on the Start button to open the Tech Essentials menu and select Control Panel. Alternatively, you can click on the Start button to open the Metro/Modern UI interface, start typing **control panel**, and select Control Panel from the Search results.

Administrative Tools is still an important part of Windows 10, a set of utilities piled together as a single Control Panel applet. You have the same options for accessing Administrative Tools in Windows 10 as listed in the prior section for Windows 8/8.1.

Settings App The Windows 10 *Settings app* combines a huge number of otherwise disparate utilities, apps, and tools traditionally spread out all over your computer into one fairly unified, handy Windows app (see Figure 3-52). Since the Settings app was introduced in Windows 8, it has taken over more and more tasks from the Control Panel. Expect it to grow as Windows 10 matures.

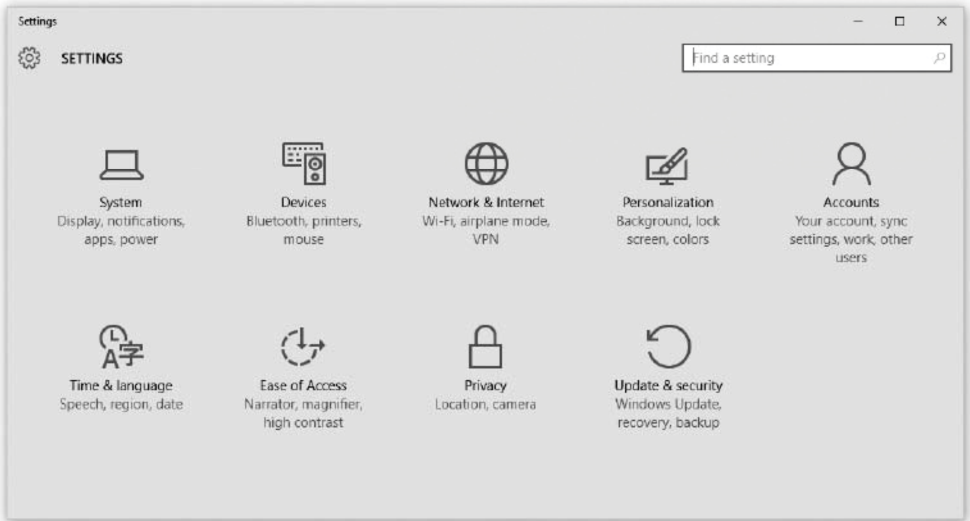
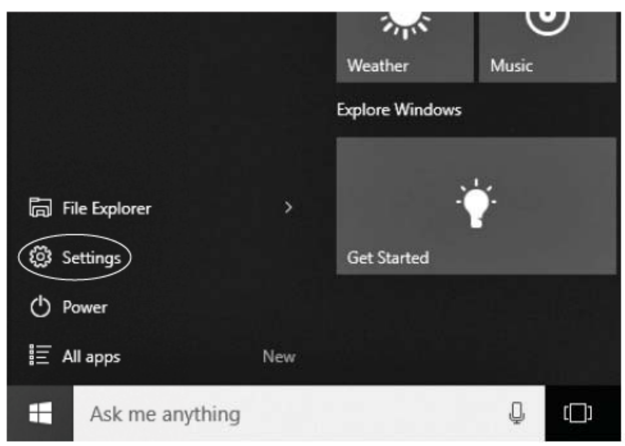


Figure 3-52 Windows Settings app

Figure 3-53 Accessing Settings in Windows 10

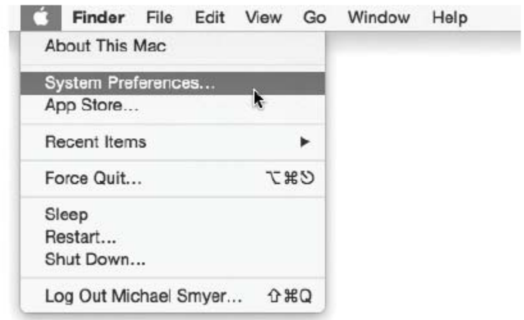


To access the Setting app, press the WINDOWS KEY to access the Start menu. Select Settings from the lower left to open the tool (see Figure 3-53).

Mac OS X

Mac OS X has two key launch points for techs: the System Preferences app and the Utilities folder. You can access both quickly.

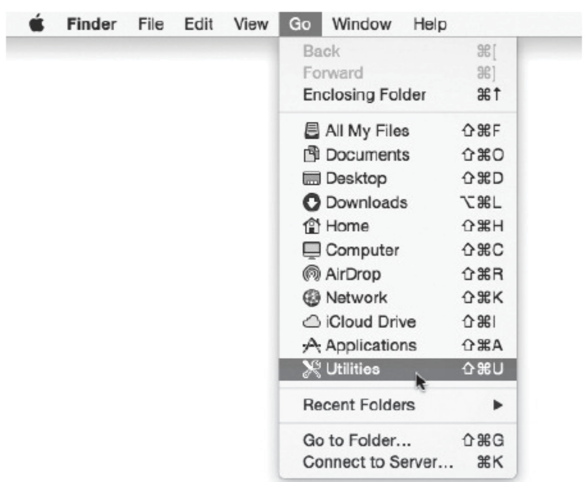
Figure 3-54 Accessing System Preferences



System Preferences To access *System Preferences*, click on the Apple (top-left corner of screen). Select System Preferences from the permanent Apple menu to open the app (see Figure 3-54). From System Preferences you have access to almost all settings you will need to administer a Mac OS X system.

Utilities Folder The second launch point is the *Utilities* folder, located neatly in the Applications folder. Because of its importance, Apple provides a quick shortcut to access it. With the Finder in focus, click on Go on the menu bar and select Utilities (see Figure 3-55). Alternatively, use the hot-key combination: COMMAND-SHIFT-U. This gives you access to the tools you need to perform services on a Mac beyond what's included in System Preferences, including Activity Monitor and Terminal. The latter

Figure 3-55 Accessing the Utilities folder



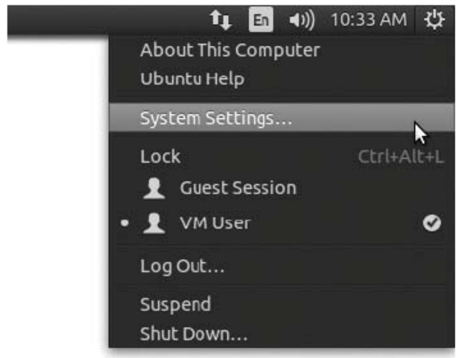


Figure 3-56 Accessing System Settings

is the command-line interface for Mac OS X, a very powerful tool for techs that we explore in detail in Chapter 16.

Linux

An essential tool in Linux for techs is the command line, called Terminal. You can get there in most distros by pressing CTRL-ALT-T. (See Chapter 16, "The Command-Line Interface," for a lot of details about essential Linux commands.)

Other launch points vary from distro to distro. Here are the locations of the launch points for the three most common desktop environments.

Unity (Default for Ubuntu Desktop) Similar to Mac OS X, Unity has a central application for managing common settings called *System Settings*. To access System Settings, click on the gear icon on the far right of the menu bar and select System Settings (see Figure 3-56).

You can find settings and utilities not in the System Settings application with the rest of the applications in the Dash. Click on the Ubuntu button at the top of the Launcher (see Figure 3-57). From here you can search or browse for handy applications such as the System Monitor or the always critical Terminal.

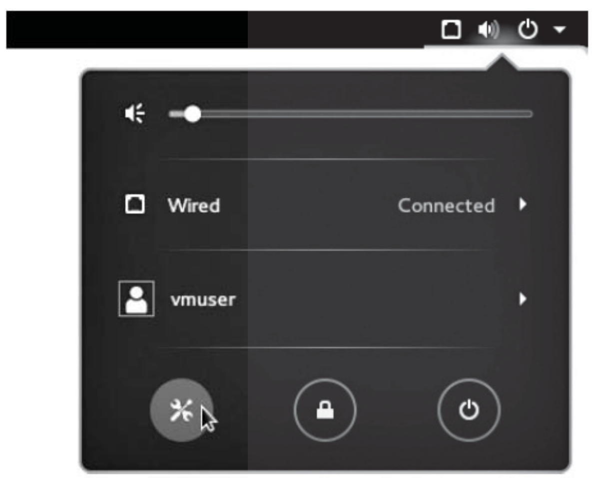
GNOME 3 (Default for Fedora Workstation, Red Hat Enterprise Linux) If you have any experience with Ubuntu's Unity, working with GNOME 3 should feel somewhat familiar because Unity is based on GNOME. Because of this connection, the same applications are used to administer GNOME 3–based desktops, although some of the names are different.

The first launch point is the All Settings application, which is practically the same as System Settings in Unity. To access All Settings, click on the down arrow icon on the far right of the menu bar and select the wrench and screwdriver icon (see Figure 3-58).

For other system utilities such as System Monitor or Terminal, click on the Activities button on the far left of the menu bar. From here you can search for the utility from the box at the top, or select the Show Applications grid icon from the bottom of the Dash on the left side of the screen. This will open a menu showing all installed applications, and within this list is a folder for Utilities.

Figure 3-57 Browsing through Dash applications

Figure 3-58 Accessing All Settings



KDE Plasma Desktop (Default for OpenSUSE, Kubuntu)

"Choice!" could be the unofficial motto of Linux, and when you are working on a KDE-based distro, you are certainly spoiled for choices. The downside to this abundance is that the configuration utilities can vary among the different KDE-based distros.

The one thing that is the same in all the KDE-based distros is that everything you need to work on the system is accessible from the *Kickoff* menu on the far left of the

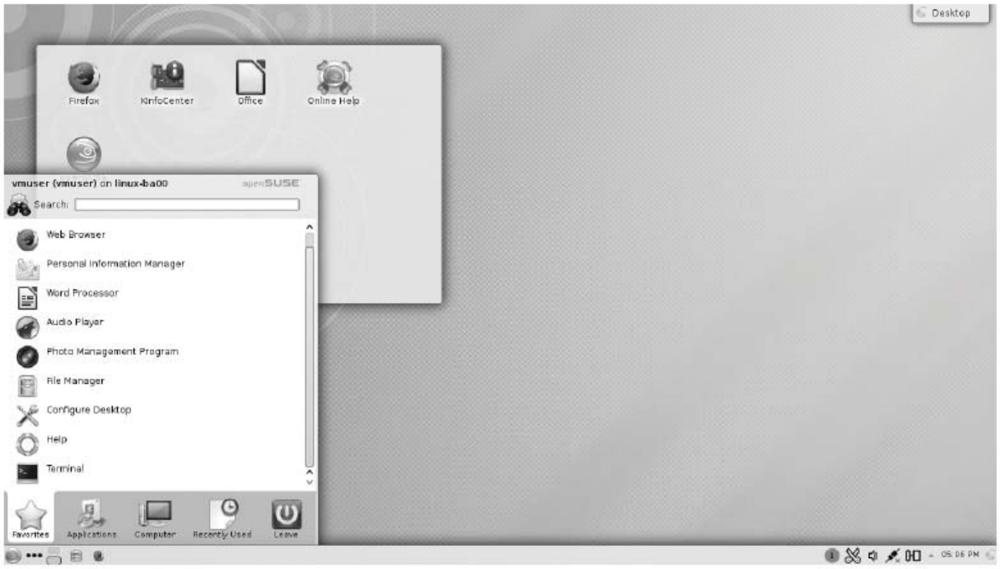


Figure 3-59 KDE Kickoff menu

Panel (see Figure 3-59). The Kickoff menu looks and works a lot like the Start menu in Windows 7, so it should be relatively easy to navigate. Once in the Kickoff menu, you can search for a needed utility or select the Applications tab at the bottom. From here, most distros have a Utilities or System menu that holds all the key system configuration and maintenance applications.

Chapter 3 Review

Questions

1. Which version of Windows introduced the Metro UI?
   - **A.** Windows 7
   - **B.** Windows 8
   - **C.** Windows 8.1
   - **D.** Windows 10
2. Which Windows 8 feature did Microsoft not include in Windows 10?
   - **A.** Metro/Modern UI
   - **B.** Start button
   - **C.** Control Panel
   - **D.** Charms bar

3. What Mac OS X feature is essentially multiple Desktops?
   A. Charms
   B. Desktop
   C. Mission Control
   D. Spaces
4. What is the default Ubuntu desktop environment?
   A. Metro UI
   B. Unity
   C. KDE
   D. GNOME 3
5. The user, Mike, has downloaded files with his Web browser. Where will they be stored by default?
   A. C:\Downloads
   B. C:\Mike\Desktop\Downloads
   C. C:\Users\Mike\Downloads
   D. C:\Users\Mike\Desktop\Downloads
6. 32-bit programs are installed into which folder by default in a 64-bit edition of Windows?
   A. C:\Program Files
   B. C:\Program Files (x32)
   C. C:\Program Files\Wins\Old
   D. C:\Program Files (x86)
7. Which Mac OS X feature is functionally equivalent to Windows File Explorer?
   A. Finder
   B. Dock
   C. Quartz
   D. File Manager
8. Which of the following paths would open Administrative Tools in Windows 8.1?
   A. Right-click the task bar and select Administrative Tools from the context menu.
   B. Right-click the Start button and select Administrative Tools from the context menu.
   C. Right-click anywhere on the desktop and select Administrative Tools from the context menu.
   D. Press the WINDOWS KEY-L combination to open Administrative Tools.

9. What feature of Mac OS X is the equivalent of the command-line interface in Windows?
   A. Dock
   B. Spaces
   C. Terminal
   D. Unity
10. What Windows app in Windows 10 combines many utilities into a unified tool?
   A. Settings
   B. Control
   C. Command Center
   D. Control Center

Answers

1. **B.** Microsoft introduced Metro UI with Windows 8.
2. **D.** Microsoft did not include the Charms bar in Windows 10. Bye!
3. **D.** *Spaces* is the term Apple uses for multiple Desktops in Mac OS X.
4. **B**. Ubuntu Linux uses the Unity DE by default.
5. **C**. The default download location in Windows is C:\Users\<user name>\Downloads.
6. **D**. By default, 32-bit applications install into the C:\Program Files (x86) folder.
7. **A**. Finder is the equivalent of File Explorer.
8. **B**. To open Administrative Tools, right-click on the Start button and select Administrative Tools. Easy!
9. **C**. Terminal is the equivalent of the Windows command-line interface.
10. **A**. The Settings app in Windows 10 offers many utilities in a unified interface.

CHAPTER 4

Microprocessors

In this chapter, you will learn how to

- Identify the core components of a CPU
- Describe the relationship of CPUs and memory
- Explain the varieties of modern CPUs
- Select and install a CPU
- Troubleshoot CPUs

The *central processing unit (CPU)* does most of the calculations that make your computer…well, a computer. The CPU, also known as a *microprocessor*, invariably hides on the motherboard below a large heat sink and often a fan assembly as well. CPU makers name their microprocessors in a fashion similar to the automobile industry: CPUs get a make and a model, such as Intel Core i7 or AMD FX-8350 Black Edition. But what's happening inside the CPU to make it able to do the amazing things asked of it every time you step up to the keyboard?

This chapter delves into microprocessors in detail. We'll first discuss how processors work and the components that enable them to interact with the rest of the computer. The second section describes how CPUs work with memory. The third section takes you on a tour of modern CPUs. The fourth section gets into practical work, selecting and installing CPUs. The final section covers troubleshooting CPUs in detail.

Historical/Conceptual

CPU Core Components

Although the computer might seem to act quite intelligently, comparing the CPU to a human brain hugely overstates its capabilities. A CPU functions more like a very powerful calculator than like a brain—but, oh, what a calculator! Today's CPUs add, subtract, multiply, divide, and move billions of numbers per second. Processing that much information so quickly makes any CPU look intelligent. It's simply the speed of the CPU, rather than actual intelligence, that enables computers to perform feats such as accessing the Internet, playing visually stunning games, or editing photos.

A good technician needs to understand some basic CPU functions to support computing devices, so let's start with an analysis of how the CPU works. If you wanted to teach someone how an automobile engine works, you would use a relatively simple example engine, right? The same principle applies here. Let's begin our study of the CPU with the granddaddy of all PC CPUs: the famous Intel 8088, invented in the late 1970s. This CPU defined the idea of the modern microprocessor and contains the same basic parts used in even the most advanced CPUs today.

The Man in the Box

Let's begin by visualizing the CPU as a man in a box (see Figure 4-1). This is one clever guy. He can perform virtually any mathematical function, manipulate data, and give answers *very quickly*.

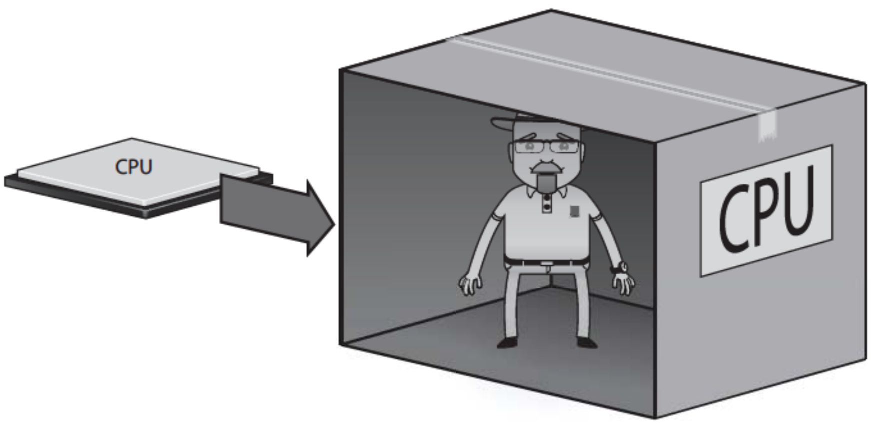


Figure 4-1 Imagine the CPU as a man in a box.

This guy is potentially very useful to us, but there's a catch—he lives closed up in a tiny box. Before he can work with us, we must come up with a way to exchange information with him (see Figure 4-2).

Imagine that we install a set of 16 light bulbs, 8 inside his box and 8 outside his box. Each of the 8 light bulbs inside the box connects to one of the 8 bulbs outside the box to form a pair. Each pair of light bulbs is always either on or off. You can control the 8 pairs of bulbs by using a set of 8 switches outside the box, and the Man in the Box can also control them by using an identical set of 8 switches inside the box. This light-bulb communication device is called the *external data bus (EDB)*.

Figure 4-3 shows a cutaway view of the external data bus. When either you or the Man in the Box flips a switch on, *both* light bulbs go on, and the switch on the other side is also flipped to the on position. If you or the Man in the Box turns a switch off, the light bulbs on both sides are turned off, along with the other switch for that pair.

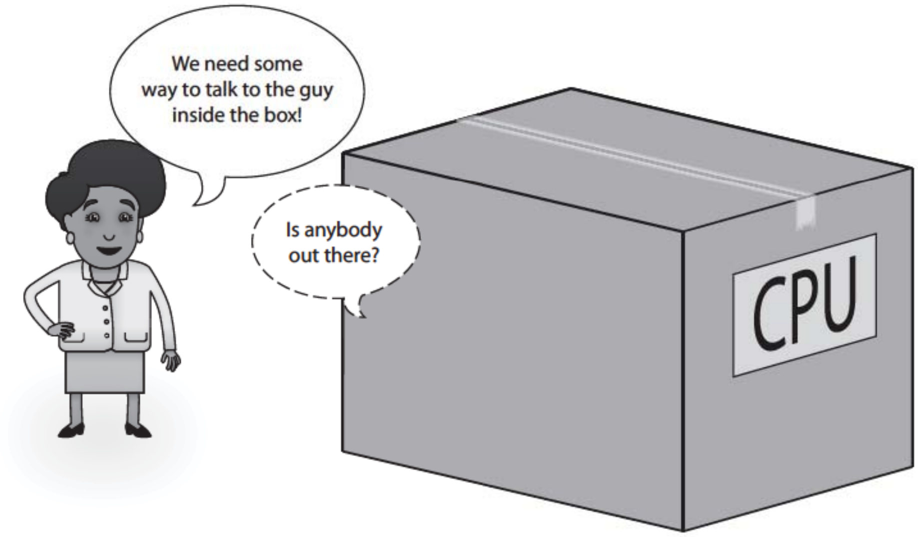


Figure 4-2 How do we talk to the Man in the Box?

Can you see how this works? By creating on/off patterns with the light bulbs that represent different pieces of data or commands, you can send that information to the Man in the Box, and he can send information back in the same way—*assuming that you agree ahead of time on what the different patterns of lights mean.* To accomplish this, you need some sort of codebook that assigns meanings to the many patterns of lights that the EDB might display. Keep this thought in mind while we push the analogy a bit more.

Before going any further, make sure you're clear on the fact that this is an analogy, not reality. There really is an EDB, but you won't see any light bulbs or switches on the CPU.

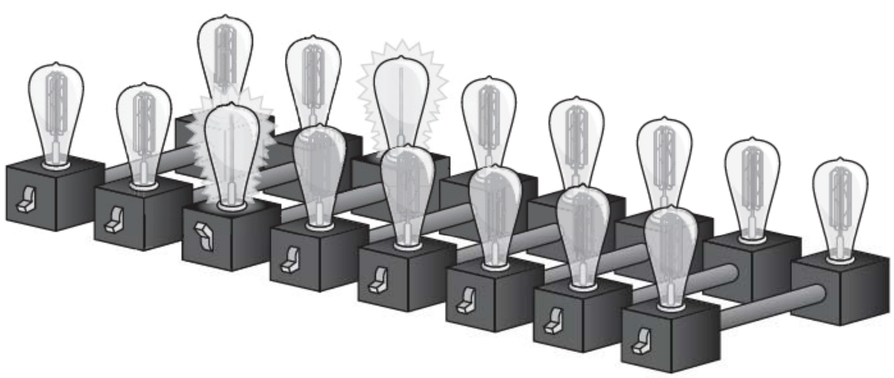


Figure 4-3 Cutaway of the external data bus—note that one light bulb pair is on.

You can, however, see little wires sticking out of many CPUs (see Figure 4-4). If you apply voltage to one of these wires, you in essence flip the switch. Get the idea? So if that wire had voltage, and if a tiny light bulb were attached to the wire, that light bulb would glow, would it not? By the same token, if the wire had no power, the light bulb would not glow. That is why the switch-and-light-bulb analogy may help you picture these little wires constantly flashing on and off.

Figure 4-4 Close-up of the underside of a CPU

Now that the EDB enables you to communicate with the Man in the Box, you need to see how it works by placing voltages on the wires. This brings up a naming problem. It's a hassle to say something like "on-off-on-off-on-on-off-off" when talking about which wires have voltage. Rather than saying that one of the EDB wires is on or off, use the number 1 to represent on and the number 0 to represent off (see Figure 4-5). That way, instead of describing the state of the lights as "on-off-on-off-on-on-off-off," I can instead describe them by writing "10101100."

In computers, wires repeatedly turn on and off. As a result, we can use this "1 and 0," or *binary*, system to describe the state of these wires at any given moment. (See, and you just thought computer geeks spoke in binary to confuse normal people. Ha!) There's much more to binary numbering in computing, but this is a great place to start.

Registers

The Man in the Box provides good insight into the workspace inside a CPU. The EDB gives you a way to communicate with the Man in the Box so you can give him work to do. But to do this work, he needs a worktable; in fact, he needs at least four worktables. Each of these four worktables has 16 light bulbs. These light bulbs are not in pairs; they're just 16 light bulbs lined up straight across the table. Each light bulb is controlled by a single switch, operated only by the Man in the Box. By creating on/off patterns like

Figure 4-5 Here "1" means on, "0" means off.

the ones on the EDB, the Man in the Box can use these four sets of light bulbs to work math problems. In a real computer, these worktables are called *registers* (see Figure 4-6) and store internal commands and data.

Registers provide the Man in the Box with a workplace for the problems you give him. All CPUs contain a large number of registers, but for the moment let's concentrate on the four most common ones: the *general-purpose registers*. Intel named them AX, BX, CX, and DX.

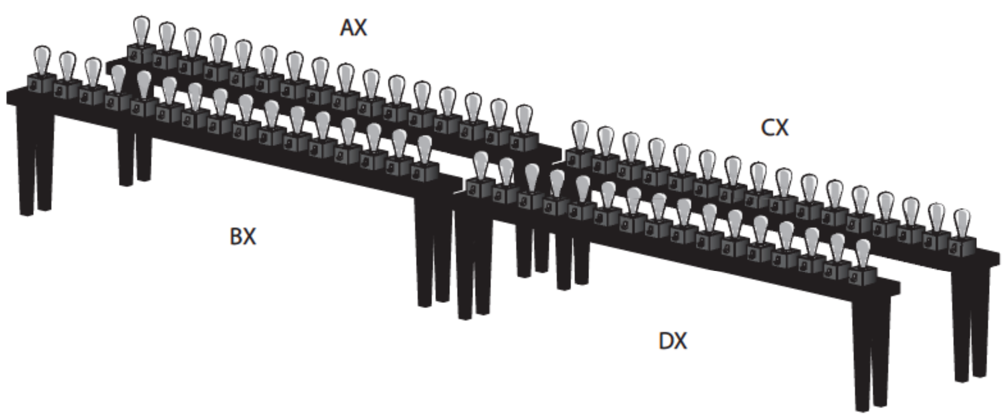


Figure 4-6 The four general-purpose registers

NOTE The 8088 was the first CPU to use the four AX–DX general-purpose registers, and they still exist in even the latest CPUs. (But they have a lot more light bulbs!) In 32-bit processors, the registers add an E for extended, so EAX, EBX, and so on. The 64-bit registers get an R for . . . I don't know, thus RAX, RBX, and so on.

Great! We're just about ready to put the Man in the Box to work, but before you close the lid on the box, you must give the Man one more tool. Remember the codebook I mentioned earlier? Let's make one to enable us to communicate with him. Figure 4-7 shows the codebook we'll use. We'll give one copy to him and make a second for us.

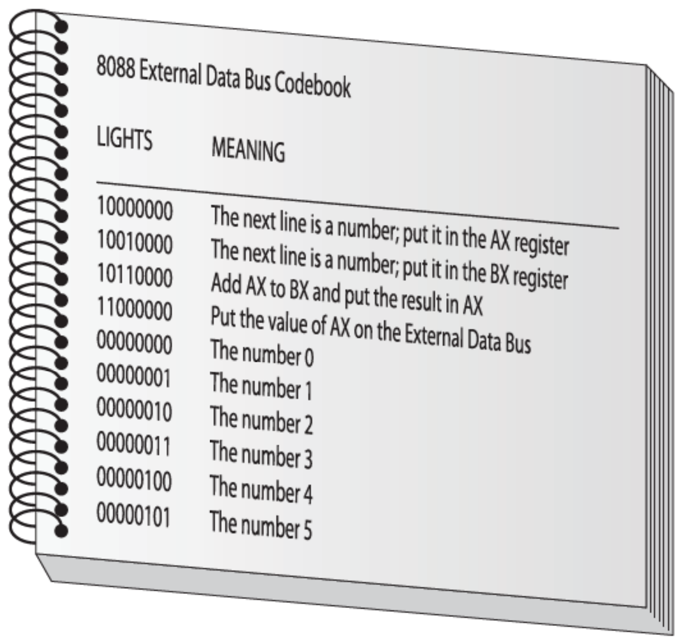


Figure 4-7
CPU codebook

In this codebook, for example, 10000111 means *Move the number 7 into the AX register*. These commands are called the microprocessor's *machine language*. The commands listed in the figure are not actual commands; as you've probably guessed, I've simplified dramatically. The Intel 8088 CPU actually used commands very similar to these, plus a few hundred others.

Here are some examples of real machine language for the Intel 8088:

10111010	The next line of code is a number. Put that number into the DX register.
01000001	Add 1 to the number already in the CX register.
00111100	Compare the value in the AX register with the next line of code.

By placing machine language commands—called *lines of code*—onto the EDB one at a time, you can instruct the Man in the Box to do specific tasks. All of the machine language commands that the CPU understands make up the CPU's *instruction set*.

So here is the CPU so far: the Man in the Box can communicate with the outside world via the EDB; he has four registers he can use to work on the problems you give him; and he has a codebook—the instruction set—so he can understand the different patterns (machine language commands) on the EDB (see Figure 4-8).

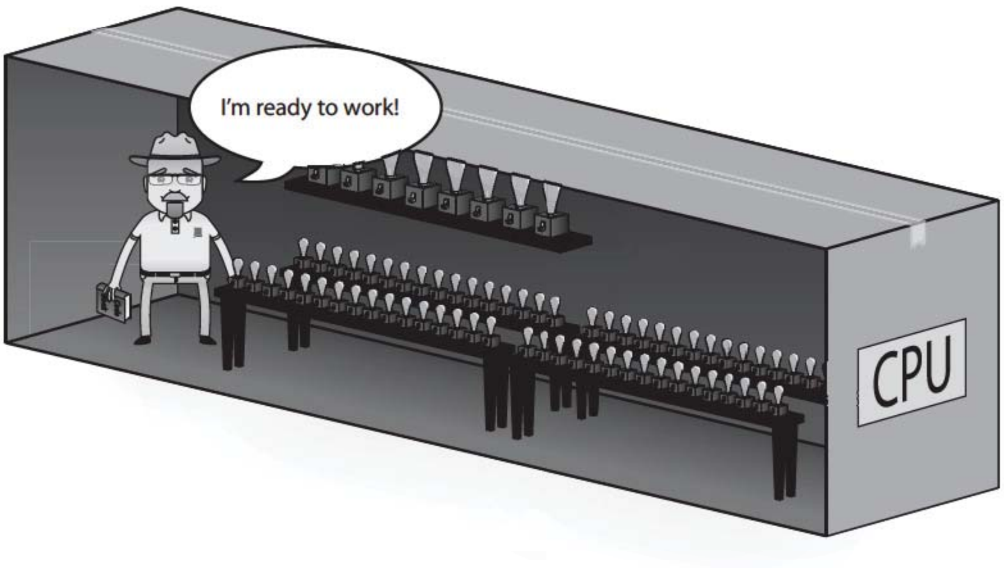


Figure 4-8 The CPU so far

Clock

Okay, so you're ready to put the Man in the Box to work. You can send the first command by lighting up wires on the EDB. How does he know when you've finished setting up the wires and it's time to act?

Have you ever seen one of those old-time manual calculators with the big crank on one side? To add two numbers, you pressed a number key, the + key, and another number key, but then to make the calculator do the calculation and give you the answer, you had to pull down the crank. That was the signal that you had finished entering data and instructions and were ready for the calculator to give you an answer.

A CPU also has a type of crank. To return to the Man in the Box, imagine there's a bell inside the box activated by a button on the outside of the box. Each time you press the button to sound the bell, the Man in the Box reads the next set of lights on the EDB. Of course, a real computer doesn't use a bell. The bell on a real CPU is a special wire called the *clock wire* (most diagrams label the clock wire CLK). A charge on the CLK wire tells the CPU that another piece of information is waiting to be processed (see Figure 4-9).

For the CPU to process a command placed on the EDB, a certain minimum voltage must be applied to the CLK wire. A single charge to the CLK wire is called a *clock cycle*. Actually, the CPU requires at least two clock cycles to act on a command, and usually

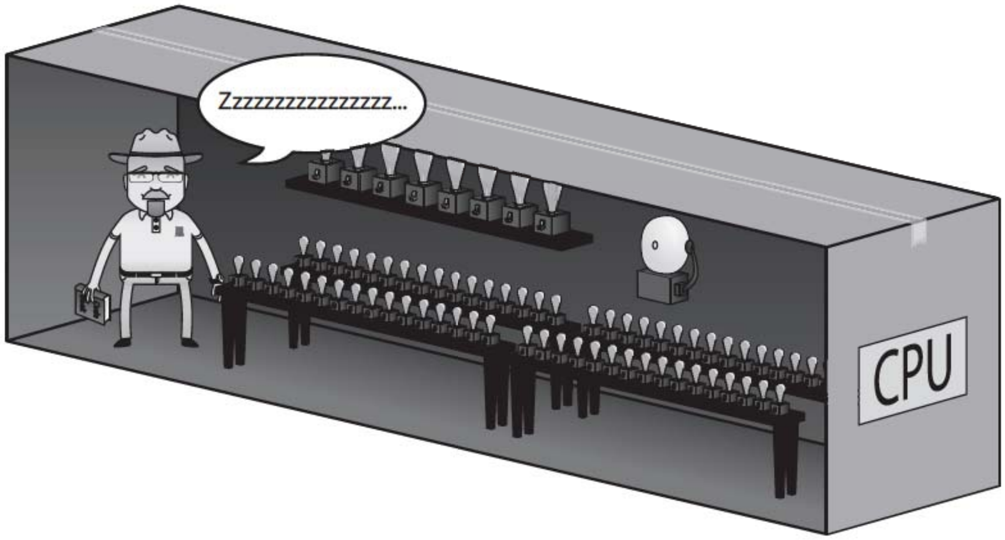


Figure 4-9 The CPU does nothing until activated by the clock.

more. Using the manual calculator analogy, you need to pull the crank at least twice before anything happens. In fact, a CPU may require hundreds of clock cycles to process some commands (see Figure 4-10).

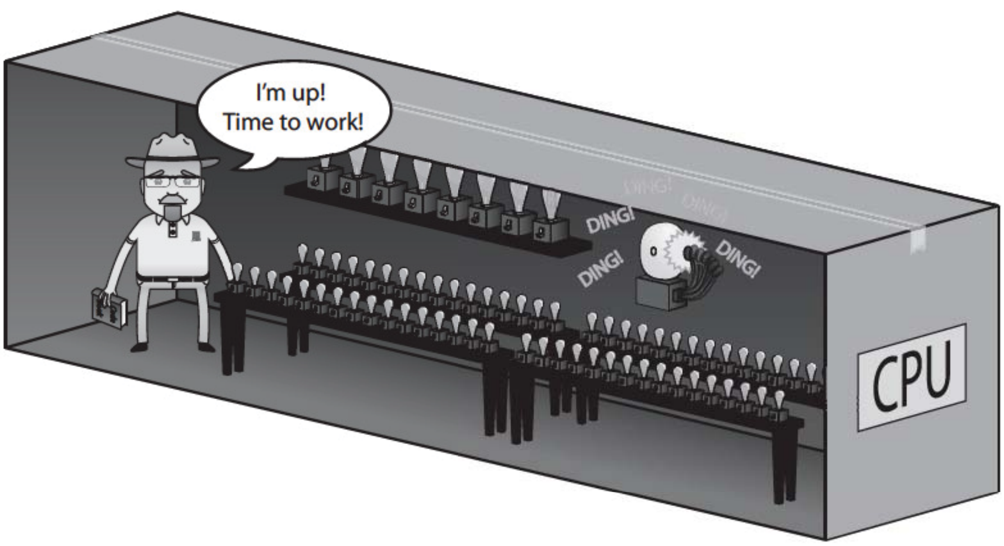


Figure 4-10 The CPU often needs more than one clock cycle to get a result.

The maximum number of clock cycles that a CPU can handle in a given period of time is referred to as its *clock speed*. The clock speed is the fastest speed at which a CPU can operate, determined by the CPU manufacturer. The Intel 8088 processor had a clock speed of 4.77 MHz (4.77 million cycles per second), extremely slow by modern standards, but still a pretty big number compared to using a pencil and paper. CPUs today run at speeds in excess of 3 GHz (3 billion cycles per second). You'll see these "hertz" terms a lot in this chapter, so here's what they mean:

1 hertz (1 Hz) = 1 cycle per second

1 megahertz (1 MHz) = 1 million cycles per second

1 gigahertz (1 GHz) = 1 billion cycles per second

A CPU's clock speed is its *maximum* speed, not the speed at which it *must* run. A CPU can run at any speed, as long as that speed does not exceed its clock speed. Manufacturers used to print the CPU's clock speed directly onto the CPU, but for the past several years they've used cryptic codes (see Figure 4-11). As the chapter progresses, you'll see why they do this.

Figure 4-11 Where is the clock speed?

The *system crystal* determines the speed at which a CPU and the rest of the PC operate. The system crystal is usually a quartz oscillator, very similar to the one in a wristwatch, soldered to the motherboard (see Figure 4-12).

NOTE CPU makers sell the exact make and model of CPU at a number of different speeds. All of these CPUs come off of the same assembly lines, so why do they have different speeds? Every CPU comes with subtle differences—flaws, really—in the silicon that makes one CPU run faster than another. The speed difference comes from testing each CPU to see what speed it can handle.

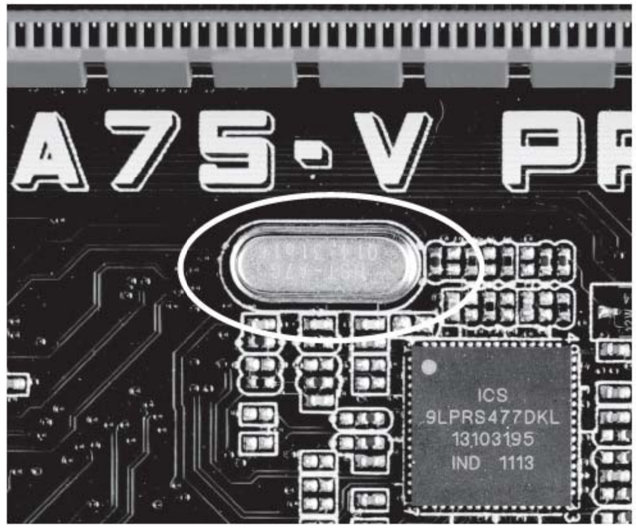


Figure 4-12 One of many types of system crystals

The quartz oscillator sends out an electric pulse at a certain speed, many millions of times per second. This signal goes first to a clock chip that adjusts the pulse, usually increasing the pulse sent by the crystal by some large multiple. (The folks who make motherboards could connect the crystal directly to the CPU's clock wire, but then if you wanted to replace your CPU with a CPU with a different clock speed, you'd need to replace the crystal too.) As long as the computer is turned on, the quartz oscillator, through the clock chip, fires a charge on the CLK wire, in essence pushing the system along.

Visualize the system crystal as a metronome for the CPU. The quartz oscillator repeatedly fires a charge on the CLK wire, setting the beat, if you will, for the CPU's activities. If the system crystal sets a beat slower than the CPU's clock speed, the CPU will work just fine, though at the slower speed of the system crystal. If the system crystal forces the CPU to run faster than its clock speed, it can overheat and stop working. Before you install a CPU into a system, you must make sure that the crystal and clock chip send out the correct clock pulse for that particular CPU. In the old days, this required very careful adjustments. With today's systems, the motherboard talks to the CPU. The CPU tells the motherboard the clock speed it needs, and the clock chip automatically adjusts for the CPU, making this process now invisible.

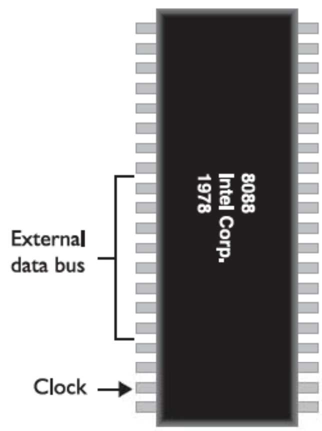
NOTE Aggressive users sometimes intentionally overclock CPUs by telling the clock chip to multiply the pulse faster than the CPU's designed speed. They do this to make slower (cheaper) CPUs run faster and to get more performance in demanding programs. See the "Overclocking" section, later in this chapter.

Back to the External Data Bus

One more reality check. We've been talking about tables with racks of light bulbs, but of course real CPU registers don't use light bulbs to represent on/1 and off/0. Registers are tiny storage areas on the CPU made up of microscopic semiconductor circuits that hold charges. It's just easier to imagine a light bulb lit up to represent a circuit holding a charge; when the light bulb is off, there is no charge.

Figure 4-13 is a diagram of an 8088 CPU, showing the wires that comprise the external data bus and the single clock wire. Because the registers are inside the CPU, you can't see them in this figure.

Figure 4-13 Diagram of an Intel 8088 showing the external data bus and clock wires



Now that you have learned what components are involved in the process, try the following simple exercise to see how the process works. In this example, you tell the CPU to add 2 + 3. To do this, you must send a series of commands to the CPU; the CPU will act on each command, eventually giving you an answer. Refer to the codebook in Figure 4-7 to translate the instructions you're giving the Man in the Box into binary commands.

Did you try it? Here's how it works:

1. Place 10000000 on the external data bus (EDB).
2. Place 00000010 on the EDB.
3. Place 10010000 on the EDB.
4. Place 00000011 on the EDB.
5. Place 10110000 on the EDB.
6. Place 11000000 on the EDB.

When you finish step 6, the value on the EDB will be 00000101, the decimal number *5* written in binary.

Congrats! You just added 2 + 3 by using individual commands from the codebook. This set of commands is known as a *program*, which is a series of commands sent to a

CPU in a specific order for the CPU to perform work. Each discrete setting of the EDB is a line of code. This program, therefore, has six lines of code.

Memory

Now that you've seen how the CPU executes program code, let's work backward in the process for a moment and think about how the program code gets to the external data bus. The program itself is stored on the hard drive. In theory, you could build a computer that sends data from the hard drive directly to the CPU, but there's a problem—the hard drive is too slow. Even the ancient 8088, with its clock speed of 4.77 MHz, could conceivably process several million lines of code every second. Modern CPUs crank out billions of lines every second. Hard drives simply can't give the data to the CPU at a fast enough speed.

Computers need some other device that takes copies of programs from the hard drive and then sends them, one line at a time, to the CPU quickly enough to keep up with its demands. Because each line of code is nothing more than a pattern of eight ones and zeros, any device that can store ones and zeros eight-across will do. Devices that in any way hold ones and zeros that the CPU accesses are known generically as *memory*.

Many types of devices store ones and zeros perfectly well—technically even a piece of paper counts as memory—but computers need memory that does more than just store groups of eight ones and zeros. Consider this pretend program:

1. Put 2 in the AX register.
2. Put 5 in the BX register.
3. If AX is greater than BX, run line 4; otherwise, go to line 6.
4. Add 1 to the value in AX.
5. Go back to line 1.
6. Put the value of AX on the EDB.

This program has an IF statement, also called a *branch* by CPU makers. The CPU needs a way to address each line of this memory—a way for the CPU to say to the memory, "Give me the next line of code" or "Give me line 6." Addressing memory takes care of another problem: the memory must store not only programs, but also the result of the programs. If the CPU adds 2 + 3 and gets 5, the memory needs to store that 5 in such a way that other programs may later read that 5, or possibly even store that 5 on a hard drive. By addressing each line of memory, other programs will know where to find the data.

Memory and RAM

Memory must store not only programs, but also data. The CPU needs to be able to read and write to this storage medium. Additionally, this system must enable the CPU to jump to *any* line of stored code as easily as to any other line of code. All of this must be done at or at least near the clock speed of the CPU. Fortunately, this magical device has existed for many years: *random access memory (RAM)*. Chapter 5 develops the concept of

RAM in detail, so for now let's look at RAM as an electronic spreadsheet, like one you can generate in Microsoft Excel (see Figure 4-14). Each cell in this spreadsheet can store only a one or a zero. Each cell is called a *bit*. Each row in the spreadsheet is 8 bits across to match the EDB of the 8088. Each row of 8 bits is called a *byte*. In PCs, RAM transfers and stores data to and from the CPU in byte-sized chunks. RAM is therefore arranged in byte-sized rows. Here are the terms used to talk about quantities of bits:

- Any individual 1 or 0 = a bit
- 4 bits = a nibble
- 8 bits = a byte
- 16 bits = a word
- 32 bits = a double word
- 64 bits = a paragraph or quad word

Figure 4-14
RAM as a spreadsheet

1	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0
0	0	0	0	1	1	0	1
0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	1
0	1	0	1	1	0	1	0
0	0	1	1	1	1	0	0
0	0	0	0	1	0	0	1
1	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0

The number of bytes of RAM varies from PC to PC. In earlier PCs, from around 1980 to 1990, the typical system would have only a few hundred thousand bytes of RAM. Today's systems often have billions of bytes of RAM.

Let's stop here for a quick reality check. Electronically, RAM looks like a spreadsheet, but real RAM is made of groups of semiconductor chips soldered onto small cards that snap into your computer (see Figure 4-15). In Chapter 5, you'll see how these groups of chips actually make themselves look like a spreadsheet. For now, don't worry about real RAM and just stick with the spreadsheet idea.

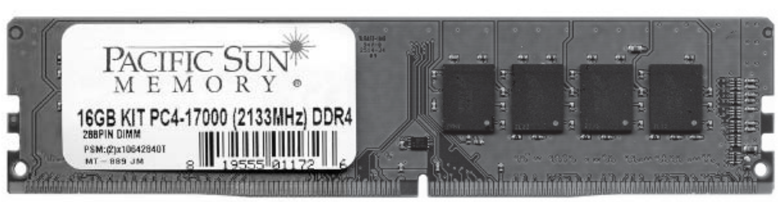


Figure 4-15
Typical RAM

The CPU accesses any one row of RAM as easily and as fast as any other row, which explains the "random access" part of RAM. Not only is RAM randomly accessible, it's also fast. By storing programs on RAM, the CPU can access and run them very quickly. RAM also stores any data that the CPU actively uses.

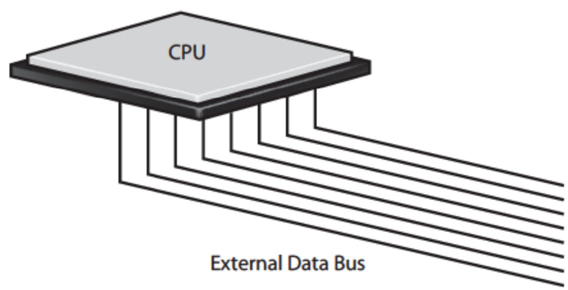
Computers use *dynamic RAM (DRAM)* for the main system memory. DRAM needs both a constant electrical charge and a periodic refresh of the circuits; otherwise, it loses data—that's what makes it dynamic rather than static in content. The refresh can cause some delays, because the CPU has to wait for the refresh to happen, but modern CPU manufacturers have clever ways to get by this issue, as you'll see when you read about modern processor technology later in this chapter.

Don't confuse RAM with mass storage devices such as hard drives and flash drives. You use hard drives and flash drives to store programs and data permanently. Chapters 9–11 discuss permanent storage in intimate detail.

Address Bus

So far, the entire PC consists of only a CPU and RAM. But the CPU and the RAM need some connection so they can talk to each other. To do so, extend the external data bus from the CPU so it can talk to the RAM (see Figure 4-16).

Figure 4-16 Extending the EDB



Wait a minute. This is not a matter of just plugging the RAM into the EDB wires! RAM is a spreadsheet with thousands and thousands of discrete rows, and you need to look at the contents of only one row of the spreadsheet at a time, right? So how do you connect the RAM to the EDB in such a way that the CPU can see any one given row but still give the CPU the capability to look at *any* row in RAM? We need some type of chip between the RAM and the CPU to make the connection. The CPU needs to be able to say which row of RAM it wants, and the chip should handle the mechanics of retrieving that row of data from the RAM and putting it on the EDB. Wouldn't you know I just happen to have such a chip? This chip comes with many names, but for right now just call it the *memory controller chip (MCC)*.

The MCC contains special circuitry so it can grab the contents of any single line of RAM and place that data or command on the EDB. This in turn enables the CPU to act on that code (see Figure 4-17).

Once the MCC is in place to grab any discrete byte of RAM, the CPU needs to be able to tell the MCC which line of code it needs. The CPU therefore gains a second set

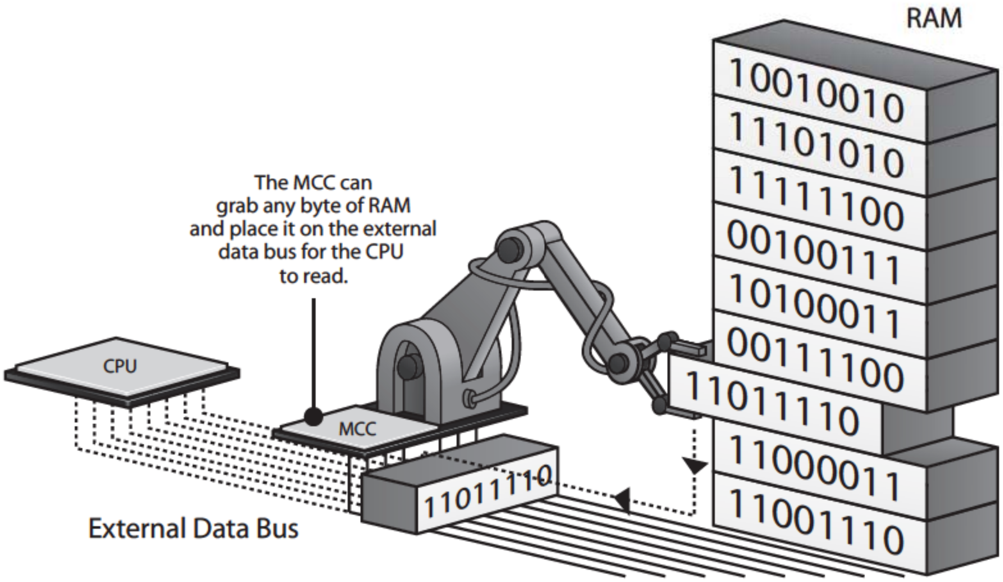


Figure 4-17 The MCC grabs a byte of RAM.

of wires, called the *address bus*, with which it can communicate with the MCC. Different CPUs have different numbers of wires (which, you will soon see, is very significant). The 8088 had 20 wires in its address bus (see Figure 4-18).

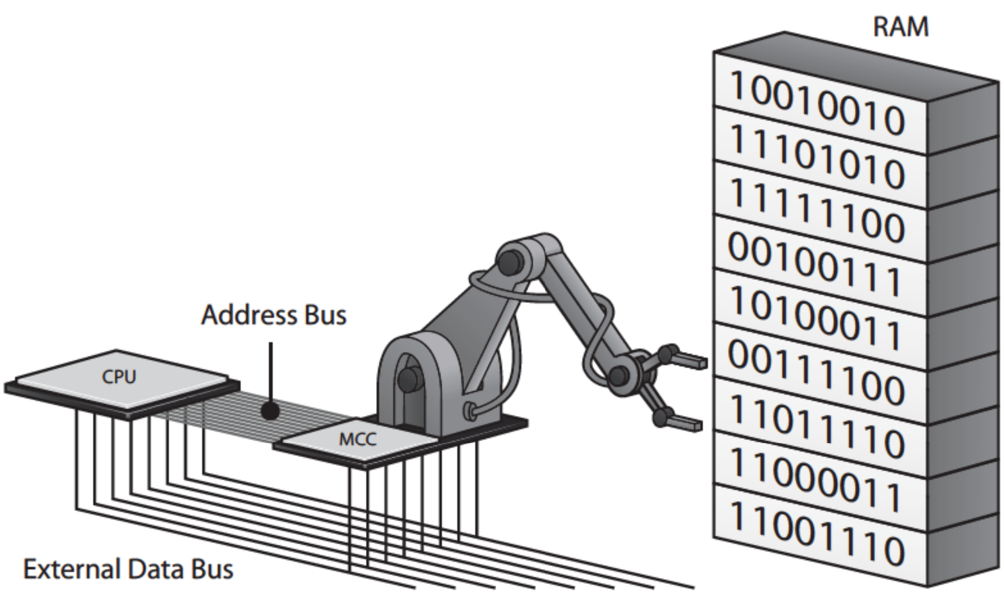


Figure 4-18 Address bus

By turning the address bus wires on and off in different patterns, the CPU tells the MCC which line of RAM it wants at any given moment. Every different pattern of ones and zeros on these 20 wires points to one byte of RAM. There are two big questions here. First, how many different patterns of on-and-off wires can exist with 20 wires? And second, which pattern goes to which row of RAM?

How Many Patterns?

Mathematics can answer the first question. Each wire in the address bus exists in only one of two states: on or off. If the address bus consisted of only one wire, that wire would at any given moment be either on or off. Mathematically, that gives you (pull out your old pre-algebra books) $2^1 = 2$ different combinations. If you have two address bus wires, the address bus wires create $2^2 = 4$ different combinations. If you have 20 wires, you would have 2^{20} (or 1,048,576) combinations. Because each pattern points to one line of code and each line of RAM is one byte, if you know the number of wires in the CPU's address bus, you know the maximum amount of RAM that a particular CPU can handle.

Because the 8088 had a 20-wire address bus, the most RAM it could handle was 2^{20}, or 1,048,576, bytes. The 8088, therefore, had an *address space* of 1,048,576 bytes. This is not to say that every computer with an 8088 CPU had 1,048,576 bytes of RAM. Far from it! The original IBM PC only had a measly 64 kilobytes—but that was considered plenty back in the Dark Ages of Computing in the early 1980s.

Okay, so you know that the 8088 had 20 address wires and a total address space of 1,048,576 bytes. Although this is accurate, no one uses such an exact term to discuss the address space of the 8088. Instead, you say that the 8088 had one *megabyte* (1 MB) of address space.

What's a "mega"? Well, let's get some terminology down. Dealing with computers means constantly dealing with the number of patterns a set of wires can handle. Certain powers of 2 have names used a lot in computing. The following list explains.

1 kilo = 2^{10} = 1024 (abbreviated as "K")

1 kilobyte = 1024 bytes (abbreviated as "KB")

1 mega = 2^{20} = 1,048,576 (abbreviated as "M")

1 megabyte = 1,048,576 bytes (abbreviated as "MB")

1 giga = 2^{30} = 1,073,741,824 (abbreviated as "G")

1 gigabyte = 1,073,741,824 bytes (abbreviated as "GB")

1 tera = 2^{40} = 1,099,511,627,776 (abbreviated as "T")

1 terabyte = 1,099,511,627,776 bytes (abbreviated as "TB")

1 peta = 2^{50} = 1,125,899,906,842,624 (abbreviated as "P")

1 petabyte = 1,125,899,906,842,624 bytes (abbreviated as "PB")

1 kilo is *not* equal to 1000 (one thousand)

1 mega is *not* equal to 1,000,000 (one million)

1 giga is *not* equal to 1,000,000,000 (one billion)

1 tera is *not* equal to 1,000,000,000,000 (one trillion)

1 peta is *not* equal to 1,000,000,000,000,000 (one quadrillion)

(But they are pretty close!)

NOTE Of course, 1 kilo is equal to 1000 when you talk in terms of the metric system. It also means 1000 when you talk about the clock speed of a chip, so 1 KHz is equal to 1000 Hz. When you talk storage capacity, though, the binary numbers kick in, making 1 KB = 1024 bytes. Got it? This same bizarre dual meaning applies all the way up the food chain, so 1 MHz is 1,000,000 Hz, but 1 MB is 1,048,576 bytes; 1 GHz is 1 billion Hz, but 1 GB is 1,073,741,824 bytes; and so on.

Some techs and standards bodies use slightly different prefixes for the metric and binary numbering systems. In such a system, 1 K is a kilo, and always means 1000. So 1 kilobyte (KB) is 1000 bytes. That's the metric value. To go binary, you'd use kibibytes (KiB). 1 KiB = 1024 bytes. The industry standard is the one outlined in this chapter, however, and that's what you'll see on the CompTIA A+ exams.

Which Pattern Goes to Which Row?

The second question is a little harder: "Which pattern goes to which row of RAM?" To understand this, let's take a moment to discuss binary counting. In binary, only two numbers exist, 0 and 1, which makes binary a handy way to work with wires that turn on and off. Let's try to count in binary: 0, 1…what's next? It's not 2—you can only use zeros and ones. The next number after 1 is 10! Now let's count in binary to 1000: 0, 1, 10, 11, 100, 101, 110, 111, 1000. Try counting to 10000. Don't worry; it hardly takes any time at all.

Super; you now count in binary as well as any math professor. Let's add to the concept. Stop thinking about binary for just a moment and think about good old base 10 (regular numbers). If you have the number 365, can you put zeros in front of the 365, like this: 000365? Sure you can—it doesn't change the value at all. The same thing is true in binary. Putting zeros in front of a value doesn't change a thing! Let's count again to 1000 in binary. In this case, add enough zeros to make 20 places:

00000000000000000000

00000000000000000001

00000000000000000010

00000000000000000011

00000000000000000100

00000000000000000101

00000000000000000110

00000000000000000111

00000000000000001000

Hey, wouldn't this be a great way to represent each line of RAM on the address bus? The CPU identifies the first byte of RAM on the address bus with 00000000000000000000. The CPU identifies the last RAM row with 11111111111111111111. When the CPU turns off all of the address bus wires, it wants the first line of RAM; when it turns on all of the wires, it wants the 1,048,576th line of RAM. Obviously, the address bus also addresses all of the rows of RAM in between. So, by lighting up different patterns of ones and zeros on the address bus, the CPU can access any row of RAM it needs.

NOTE Bits and bytes are abbreviated differently. Bits get a lowercase b, whereas bytes get a capital B. So, for example, 4 Kb is four kilobits, but 4 KB is four kilobytes. The big-B little-b standard applies all the way up the food chain, so 2 Mb = 2 megabits; 2 MB = 2 megabytes; 4 Gb = 4 gigabits; 4 GB = 4 gigabytes; and so on.

901

Modern CPUs

CPU manufacturers have achieved stunning progress with microprocessors since the days of the Intel 8088, and the rate of change doesn't show any signs of slowing. At the core, though, today's CPUs function similarly to the processors of your forefathers. The arithmetic logic unit (ALU)—that's the Man in the Box—still crunches numbers many millions of times per second. CPUs rely on memory to feed them lines of programming as quickly as possible.

This section brings the CPU into the present. We'll first look at models you can buy today, and then we'll turn to essential improvements in technology you should understand.

Developers

When IBM awarded Intel the contract to provide the CPUs for its new IBM PC back in 1980, it established for Intel a virtual monopoly on all PC CPUs. The other home-computer CPU makers of the time faded away: MOS Technology, Zilog, Motorola—no one could compete directly with Intel. Over time, other competitors have risen to challenge Intel's market-segment share dominance. In particular, a company called Advanced Micro Devices (AMD) began to make clones of Intel CPUs, creating an interesting and rather cutthroat competition with Intel that lasts to this day.

NOTE The ever-growing selection of mobile devices, such as the Apple iPhone and iPad, use a CPU architecture developed by ARM Holdings, called *ARM*. ARM-based processors use a simpler, more energy-efficient design, the reduced instruction set computing (RISC) architecture. They're not as raw powerful as the Intel and AMD complex instruction set computing (CISC) chips, but the savings in cost and battery life make ARM-based processors ideal for mobile devices.

(Note that the clear distinction between RISC and CISC processors has blurred. Each design today borrows features of the other design to increase efficiency.)

ARM Holdings designs ARM CPUs but doesn't manufacture them. Many other companies—most notably, Samsung—license the design and manufacture their own versions. Chapter 25 goes into more detail on ARM processors.

Intel

Intel Corporation thoroughly dominated the personal computer market with its CPUs and motherboard support chips. At nearly every step in the evolution of the PC, Intel has led the way with technological advances and surprising flexibility for such a huge corporation. Intel CPUs—and more specifically, their instruction sets—define the personal computer. Intel currently produces a dozen or so models of CPU for both desktop and portable computers. Most of Intel's desktop and laptop processors are sold under the Core, Pentium, and Celeron brands. Their very low-power portable/smartphone chips are branded Atom; their high-end server chips are called Xeon.

AMD

You can't really talk about CPUs without mentioning Advanced Micro Devices. AMD makes superb CPUs for the PC market and provides competition that keeps Intel on its toes. Like Intel, AMD doesn't just make CPUs, but their CPU business is certainly the part that the public notices. AMD has made CPUs that clone the function of Intel CPUs. If Intel invented the CPU used in the original IBM PC, how could AMD make clone CPUs without getting sued? Chipmakers have a habit of exchanging technologies through cross-license agreements. Way back in 1976, AMD and Intel signed just such an agreement, giving AMD the right to copy certain types of CPUs.

The trouble started with the Intel 8088. Intel needed AMD's help to supply enough CPUs to satisfy IBM's demands. But after a few years, Intel had grown tremendously and no longer wanted AMD to make CPUs. AMD said, "Too bad. See this agreement you signed?" Throughout the 1980s and into the 1990s, AMD made pin-for-pin identical

CPUs that matched the Intel lines of CPUs (see Figure 4-19). You could yank an Intel CPU out of a system and snap in an AMD CPU—no problem!

Figure 4-19 Identical Intel and AMD 486 CPUs from the early 1990s

In January 1995, after many years of legal wrangling, Intel and AMD settled and decided to end the licensing agreements. As a result of this settlement, AMD chips are no longer compatible with sockets or motherboards made for Intel CPUs—even though in some cases the chips look similar. Today, if you want to use an AMD CPU, you must purchase a motherboard designed for AMD CPUs. If you want to use an Intel CPU, you must purchase a motherboard designed for Intel CPUs. So you now have a choice: Intel or AMD.

Model Names

Intel and AMD differentiate product lines by using different product names, and these names have changed over the years. For a long time, Intel used *Pentium* for its flagship model, just adding model numbers to show successive generations—Pentium, Pentium II, Pentium III, and so on. AMD used the *Athlon* brand in a similar fashion.

Most discussions on PC CPUs focus on four end-product lines: desktop PC, budget PC, portable PC, and server computers. Table 4-1 displays many of the current product lines and names.

Market	Intel	AMD
Mainstream and enthusiast desktop	Core i7/i5/i3	A-Series, FX
Budget desktop	Pentium, Celeron	Sempron, Athlon
Portable/Mobile	Core i7/i5/i3 (mobile), Core M, Atom	A-Series
Server	Xeon	Opteron

Table 4-1 Current Intel and AMD Product Lines and Names

Both Intel and AMD reuse model names for products aimed at different markets. Yesterday's Pentium brand used to be for the highest end, for example, but now Intel uses the brand for its budget market. The same thing happened to the Athlon brand. To add a little more confusion, the budget CPUs are not the older CPUs still being sold, but low-end versions of current model lines.

Code Names

Both Intel and AMD continue to refine the CPU manufacturing process after releasing a new model, but they try to minimize the number of model names in use. This means that they release CPUs labeled as the same model, but the CPUs inside can be very different from earlier versions of that model. Both companies use *code names* to keep track of different variations within models (see Figure 4-20). As a tech, you need to know both the models and code names to be able to make proper recommendations for your clients. One example illustrates the need: the Intel Core i7.

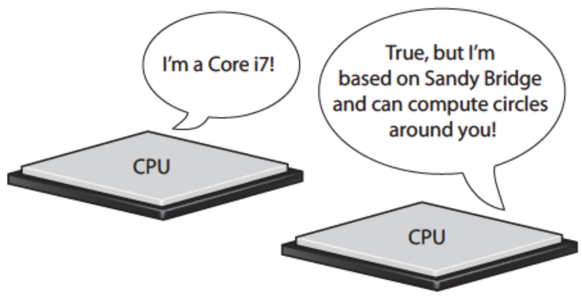


Figure 4-20 Same branding, but different capabilities

Intel released the first Core i7 in the summer of 2008. By spring of 2012, the original microarchitecture—code-named Nehalem—had gone through five variations, none of which worked on motherboards designed for one of the other variations. Plus, in 2011, Intel introduced the Sandy Bridge version of the Core i7 that eventually had two desktop versions and a mobile version, all of which used still other sockets. Just about every year since then has seen a new Core i7 based on improved architectures with different code names such as Ivy Bridge, Haswell, Broadwell, and so on. (And I'm simplifying the variations here.)

NOTE The processor number helps a lot when comparing processors once you decode the meanings. We need to cover more about modern processors before introducing processor numbers. Look for more information in the next section, "Selecting a CPU."

At this point, a lot of new techs throw their hands in the air. How do you keep up? How do you know which CPU will give your customer the best value for his or her money and provide the right computing firepower for his or her needs? Simply put, you need to research efficiently.

Your first stop should be the manufacturers' Web sites. Both companies put out a lot of information on their products.

- www.intel.com
- www.amd.com

You can also find many high-quality tech Web sites devoted to reporting on the latest CPUs. When a client needs an upgrade, surf the Web for recent articles and make comparisons. Because you'll understand the underlying technology from your CompTIA A+ studies, you'll be able to follow the conversations with confidence. Here's a list of some of the sites I use:

- www.arstechnica.com
- www.anandtech.com
- www.tomshardware.com
- www.bit-tech.net

Finally, you can find great, exhaustive articles on all things tech at Wikipedia:

- www.wikipedia.org

NOTE Wikipedia is a user-generated, self-regulated resource. I've found it to be accurate on technical issues the vast majority of the time, but you should always check other references as well. Nicely, most article authors on the site will tell you their sources through footnotes. You can often use the Wikipedia articles as jump-off points for deeper searches.

Desktop versus Mobile

Mobile devices, such as portable computers, have needs that differ from those of desktop computers, notably the need to consume as little electricity as possible. This helps in two ways: extending battery charge and creating less heat.

Both Intel and AMD have engineers devoted to making excellent mobile versions of their CPUs that sport advanced energy-saving features (see Figure 4-21). Intel's SpeedStep technology, for example, enables the CPU to run in very low power mode and scale up automatically if the user demands more power from the CPU. If you're surfing the Web at an airport terminal, the CPU doesn't draw too much power. When you switch to playing an action game, the CPU kicks into gear. Saving energy by making the CPU run more slowly when demand is light is generically called *throttling*.

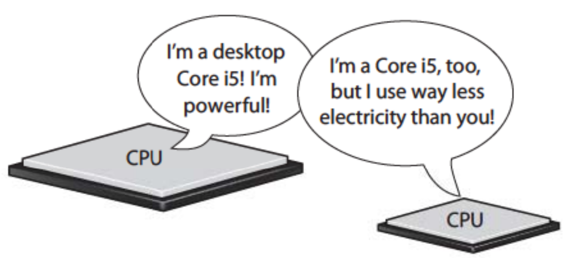


Figure 4-21 Desktop vs. mobile, fight!

Many of the technologies developed for mobile processors have migrated into their more power-hungry desktop siblings, too. That's an added bonus for the planet.

Technology

Although microprocessors today still serve the same function as the venerable 8088—crunching numbers—they do so far more efficiently. Engineers have altered, enhanced, and improved CPUs in a number of ways. This section looks at eight features:

- Clock multipliers
- 64-bit processing
- Virtualization support
- Parallel execution
- Multicore processing
- Integrated memory controller (IMC)
- Integrated graphics processing unit (GPU)
- Security

Clock Multipliers

All modern CPUs run at some multiple of the system clock speed. The system bus on my Core i7 machine, for example, runs at 100 MHz. The clock multiplier goes up to ×35 at full load to support the 3.4 GHz maximum speed. Originally, CPUs ran at the speed of the bus, but engineers early on realized the CPU was the only thing doing any work much of the time. If the engineers could speed up just the internal operations of the CPU and not anything else, they could speed up the whole computing process. Figure 4-22

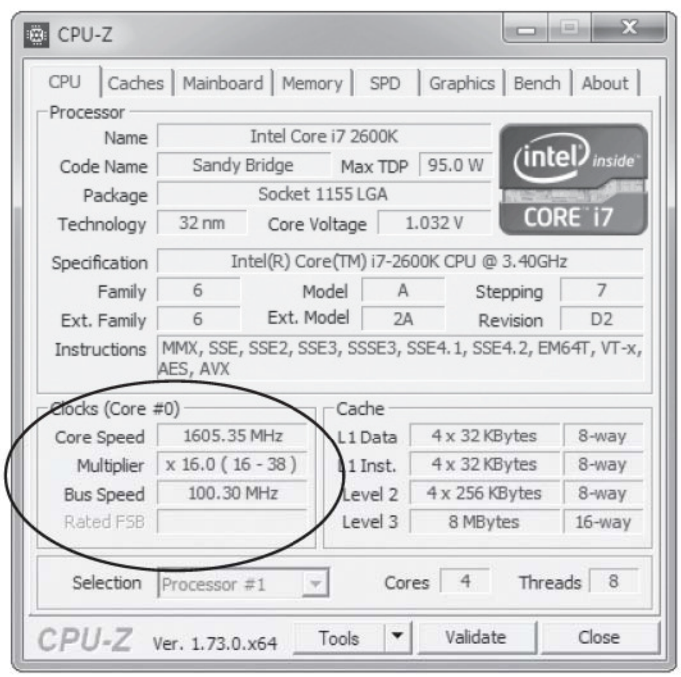


Figure 4-22 CPU-Z showing the clock speed, multiplier, and bus speed of a Core i7 processor hardly breaking a sweat

shows a nifty program called CPU-Z displaying my CPU details. Note that all I'm doing is typing at the moment, so SpeedStep has dropped the clock multiplier down to ×16 and the CPU core speed is only 1600 MHz.

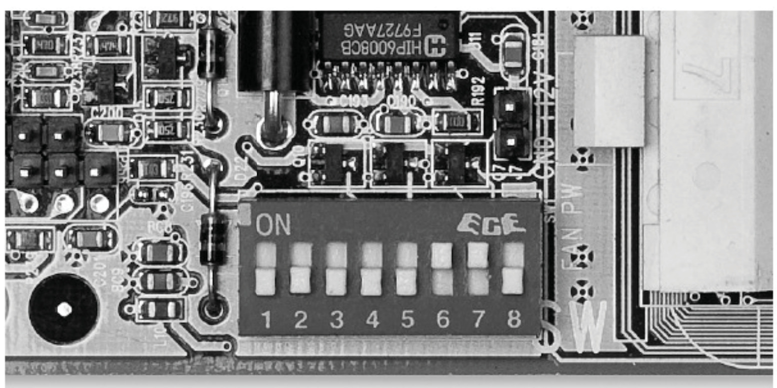
Try This!

CPU-Z

Imagine a scenario where you're dumped into an office full of unfamiliar PCs. There's no documentation about the systems at all, so your boss tells you to get cracking and find out as much as possible about each PC ASAP. Try This! Download a copy of the very popular and free CPU-Z utility from www.cpuid.com. CPU-Z gives you every piece of information you'll ever want to know about a CPU. Copy it to a thumb drive, then insert it into a bunch of different computers. (Ask permission, of course!) What kinds of processors do you find in your neighbors' computers? What can you tell about the different capabilities?

The clock speed and the multiplier on early clock-multiplying systems had to be manually configured via jumpers or dual in-line package (DIP) switches on the motherboard (see Figure 4-23). Today's CPUs report to the motherboard through a function called CPUID (CPU identifier), and the speed and multiplier are set automatically. (You can manually override this automatic setup on many motherboards. See "Overclocking," later in this chapter, for details.)

Figure 4-23 DIP switches on a motherboard



64-Bit Processing

Over successive generations of microprocessors, engineers have upgraded many physical features of CPUs. The EDB gradually increased in size, from 8- to 16- to 32- to 64-bits wide. The address bus similarly jumped, going from 20- to 24- to 32-bits wide (where it stayed for a decade).

The technological features changed as well. Engineers added new and improved registers, for example, that used fancy names like multimedia extensions (MMX) and

Streaming SIMD Extensions (SSE). A mighty shift started several years ago and continues to evolve: the move to 64-bit computing.

Most new CPUs support *64-bit processing*, meaning they can run a compatible 64-bit operating system, such as Windows 8.1, and 64-bit applications. They also support 32-bit processing for 32-bit operating systems, such as some Linux distributions, and 32-bit applications. The general-purpose registers also make the move up to 64-bit. The primary benefit to moving to 64-bit computing is that modern systems can support much more than the 4 GB of memory supported with 32-bit processing.

With a 64-bit address bus, CPUs can address 2^{64} bytes of memory, or more precisely, 18,446,744,073,709,551,616 bytes of memory—that's a lot of RAM! This number is so big that gigabytes and terabytes are no longer convenient, so we now go to an exabyte (2^{60}), abbreviated *EB*. A 64-bit address bus can address 16 EB of RAM.

In practical terms, 64-bit computing greatly enhances the performance of programs that work with large files, such as video-editing applications. You'll see a profound improvement moving from 4 GB to 8 GB or 16 GB of RAM with such programs.

x86 The terminology of CPUs can trip up new techs, so here's the scoop. CPUs from the early days can be lumped together as *x*86 CPUs, because they used an instruction set that built upon the earliest Intel CPU architecture. The Intel Core 2 Duo, for example, could run a program written for an ancient 80386 processor that was in fashion in the early 1990s.

x64 When the 64-bit CPUs went mainstream, marketing folks needed some way to mark applications, operating systems, and so on, such that consumers could quickly tell the difference between something compatible with their system or something not compatible. Since you generally cannot return software after you open it, this is a big deal. The marketing folks went with *x*64, and that created a mess.

x86-64 The earlier 32-bit stuff had been marketed as *x*86, not *x*32, so now we have *x*86 (old, 32-bit stuff) vs. *x*64 (new, 64-bit stuff). It's not pretty, but do you get the difference? To make matters even worse, however, *x*64 processors quite happily handle *x*86 code and are, by definition, *x*86 processors too! It's common to marry the two terms and describe current 64-bit CPUs as *x*86-64 processors.

Virtualization Support

Intel and AMD have built in support for running more than one operating system at a time, a process called *virtualization*. Virtualization is very cool and gets its own chapter later in the book (Chapter 18), so I'll skip the details here. The key issue from a CPU standpoint is that virtualization used to work entirely through software. Programmers had to write a ton of code to enable a CPU—which was designed to run one OS at a time—to run more than one OS at the same time. Think about the issues involved. How does the memory get allocated, for example, or how does the CPU know which OS to update when you type something or click an icon? With hardware-based virtualization support, CPUs took a lot of the burden off the programmers and made virtualization a whole lot easier.

Parallel Execution

Modern CPUs can process multiple commands and parts of commands in parallel, which is known as *parallel execution*. Early processors had to do everything in a strict, linear fashion. The CPUs accomplish this parallelism through multiple pipelines, dedicated caches, and the capability to work with multiple threads or programs at one time. To understand the mighty leap in efficiency gained from parallel execution, you need insight into the processing stages.

Pipelining To get a command from the data bus, do the calculation, and then get the answer back out on the data bus, a CPU takes at least four steps (each of these steps is called a *stage*):

1. **Fetch** Get the data from the EDB.
2. **Decode** Figure out what type of command needs to be executed.
3. **Execute** Perform the calculation.
4. **Write** Send the data back onto the EDB.

Smart, discrete circuits inside the CPU handle each of these stages. In early CPUs, when a command was placed on the data bus, each stage did its job and the CPU handed back the answer before starting the next command, requiring at least four clock cycles to process a command. In every clock cycle, three of the four circuits sat idle. Today, the circuits are organized in a conveyer-belt fashion called a *pipeline*. With pipelining, each stage does its job with each clock-cycle pulse, creating a much more efficient process. The CPU has multiple circuits doing multiple jobs, so let's add pipelining to the Man in the Box analogy. Now, it's *Men* in the Box (see Figure 4-24)!

Pipelines keep every stage of the processor busy on every click of the clock, making a CPU run more efficiently without increasing the clock speed. Note that at this point, the CPU has four stages: fetch, decode, execute, and write—a four-stage pipeline. No CPU ever made has fewer than four stages, but advancements in caching (see "Cache," next) have increased the number of stages over the years. Current CPU pipelines contain many more stages, up to 20 in some cases.

Pipelining isn't perfect. Sometimes a stage hits a complex command that requires more than one clock cycle, forcing the pipeline to stop. Your CPU tries to avoid these stops, or *pipeline stalls*. The decode stage tends to cause the most pipeline stalls; certain commands are complex and therefore harder to decode than other commands. Current processors use multiple decode stages to reduce the chance of pipeline stalls due to complex decoding.

The inside of the CPU is composed of multiple chunks of circuitry to handle the many types of calculations your PC needs to do. For example, one part, the *arithmetic logic unit (ALU)* (or *integer unit*), handles integer math: basic math for numbers with no decimal point. A perfect example of integer math is 2 + 3 = 5. The typical CPU spends most of its work doing integer math. CPUs also have special circuitry to handle complex numbers, called the *floating point unit (FPU)*. With a single pipeline, only the ALU or the FPU worked at any execution stage. Worse yet, floating point calculation often took many, many clock cycles to execute, forcing the CPU to stall the pipeline until the FPU finished executing the complex command (see Figure 4-25). Current CPUs offer multiple pipelines to keep the processing going (see Figure 4-26).

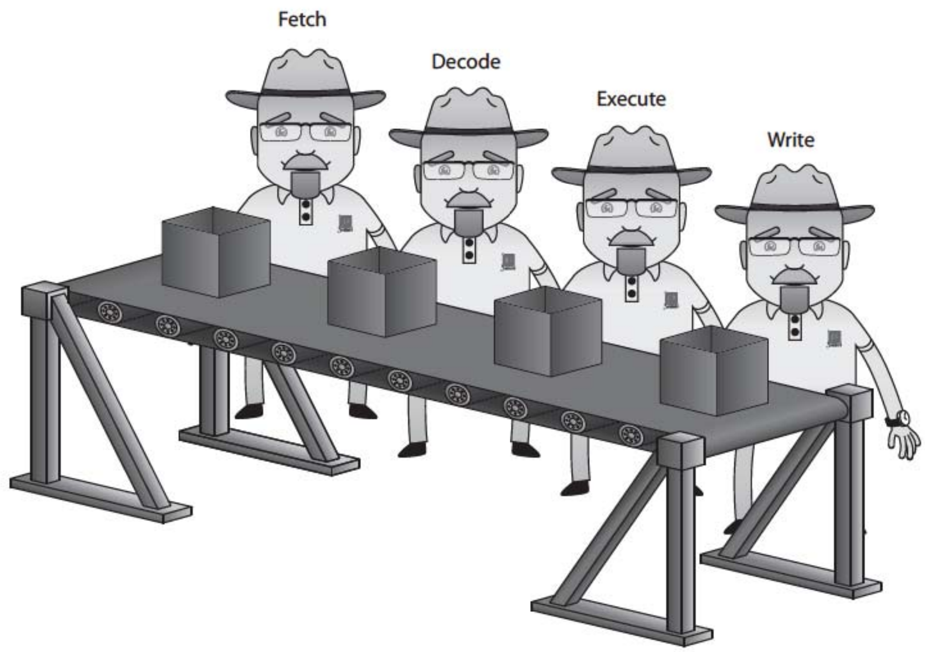


Figure 4-24 Simple pipeline

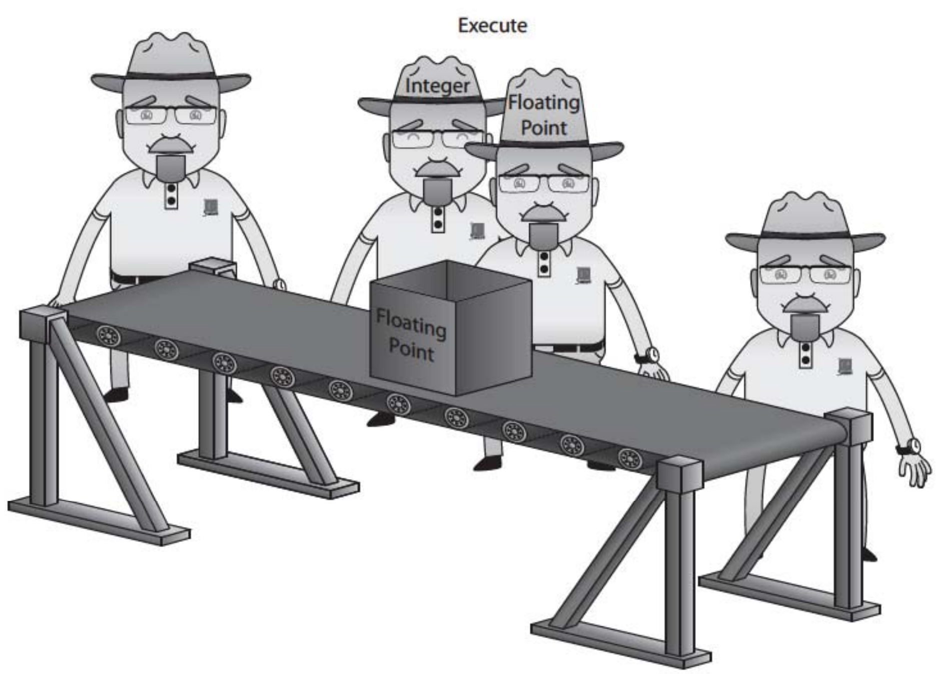


Figure 4-25 Bored integer unit

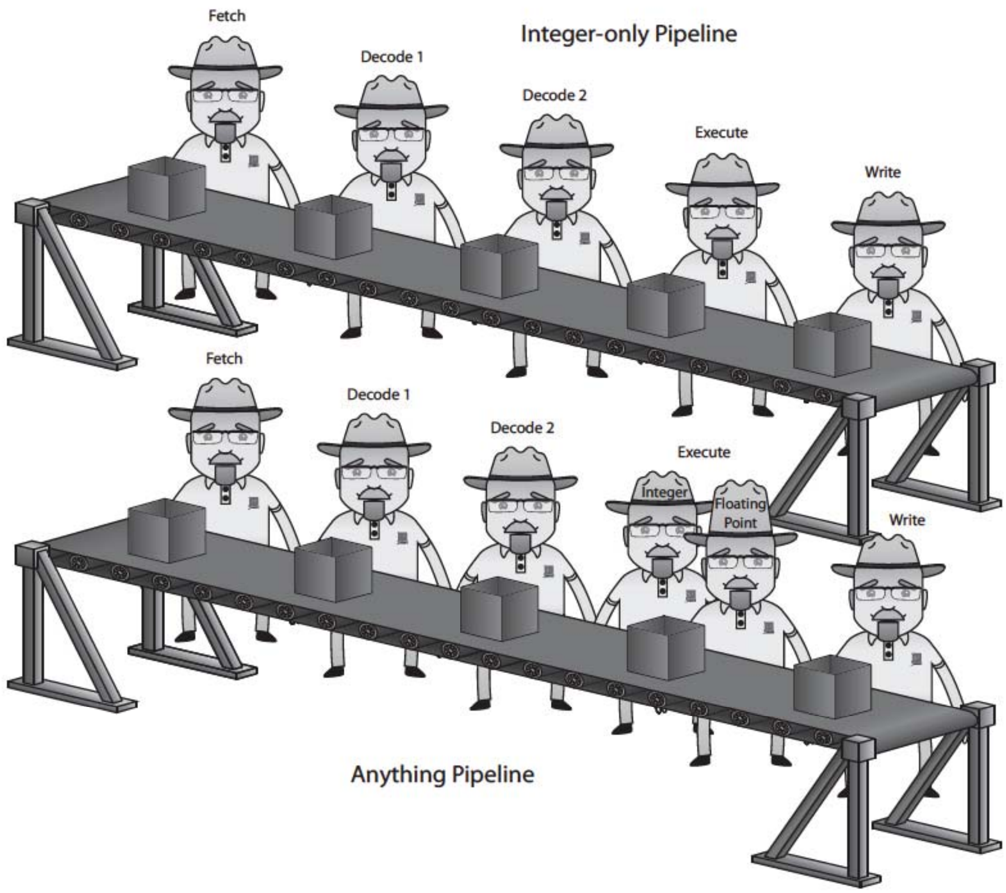


Figure 4-26 Multiple pipelines

Cache When you send a program to the CPU, you actually run lots of little programs all at the same time. Okay, let's be fair here: *you* didn't run all of these little programs—you just started your Web browser or some other program. The moment you double-clicked that icon, Windows started sending many programs to the CPU. Each of these programs breaks down into some number of little pieces, called *threads*, and data. Each thread is a series of instructions designed to do a particular job with the data.

Modern CPUs don't execute instructions sequentially—first doing step 1, then step 2, and so on—but rather process all kinds of instructions. Most applications have certain instructions and data that get reused, sometimes many times.

Pipelining CPUs work fantastically well as long as the pipelines stay filled with instructions. Because the CPU runs faster than the RAM can supply it with code, you'll always get pipeline stalls—called *wait states*—because the RAM can't keep up with the

CPU. To reduce wait states, CPUs come with built-in, very high-speed RAM called *static RAM (SRAM)*. This SRAM preloads as many instructions as possible and keeps copies of already-run instructions and data in case the CPU needs to work on them again (see Figure 4-27). SRAM used in this fashion is called a *cache*.

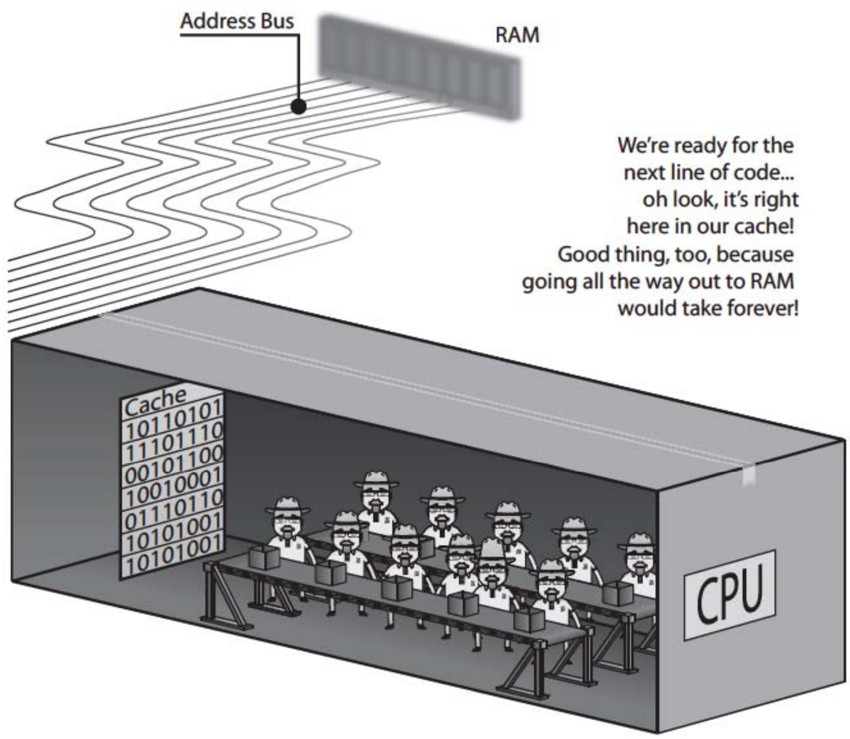


Figure 4-27 SRAM cache

The SRAM cache inside the early CPUs was tiny, only about 16 KB, but it improved performance tremendously. In fact, it helped so much that many motherboard makers began adding a cache directly to the motherboards. These caches were much larger, usually around 128 to 512 KB. When the CPU looked for a line of code, it first went to the built-in cache; if the code wasn't there, the CPU went to the cache on the motherboard. The cache on the CPU was called the *L1 cache* because it was the one the CPU first tried to use. The cache on the motherboard was called the *L2 cache*, not because it was on the motherboard, but because it was the second cache the CPU checked.

Eventually, engineers took this cache concept even further and added the L2 cache onto the CPU package. Most newer CPUs include three caches: an L1, an L2, and an L3 cache (see Figure 4-28).

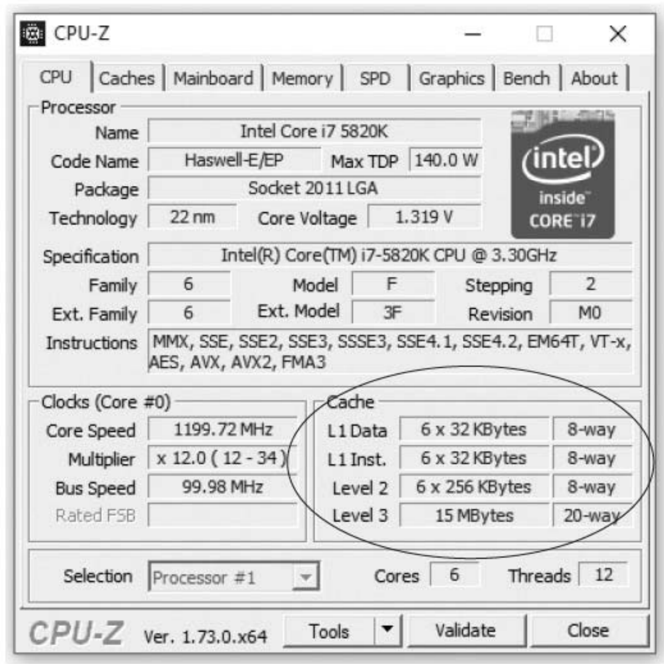


Figure 4-28 CPU-Z displaying the cache information for a Core i7 processor

The L2 cache on the early CPUs that had L2 cache included on the CPU package ran at a slower clock speed than the L1 cache. The L1 cache was in the CPU and thus ran at the speed of the CPU. The L2 cache connected to the CPU via a tiny set of wires on the CPU package. The first L2 caches ran at half the speed of the CPU.

The inclusion of the L2 cache on the chip gave rise to some new terms to describe the connections between the CPU, MCC, RAM, and L2 cache. The address bus and external data bus (connecting the CPU, MCC, and RAM) were lumped into a single term called the *frontside bus*, and the connection between the CPU and the L2 cache became known as the *backside bus* (see Figure 4-29). (These terms don't apply well to current computers, so they have fallen out of use. See the "Integrated Memory Controller" section, later in this chapter.)

NOTE To keep up with faster processors, motherboard manufacturers began to double and even quadruple the throughput of the frontside bus. Techs sometimes refer to these as *double-pumped* and *quad-pumped* frontside buses.

EXAM TIP Typically, the CompTIA A+ exams expect you to know that L1 cache will be the smallest and fastest cache; L2 will be bigger and slower than L1; and L3 will be the biggest and slowest cache. (This is not completely true anymore, with L1 and L2 running the same speed in many CPUs, but it is how it will appear on the exams.)

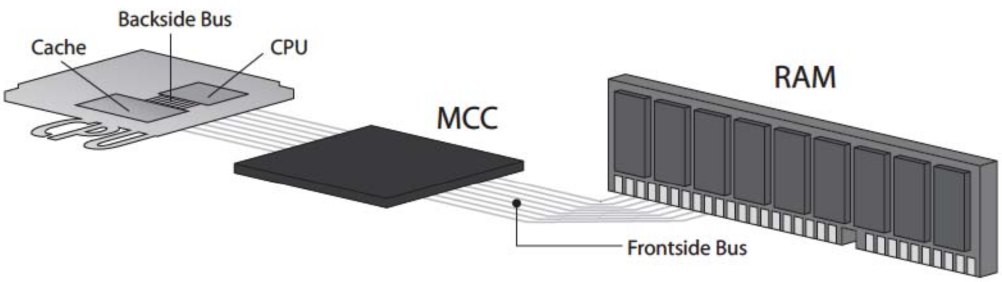
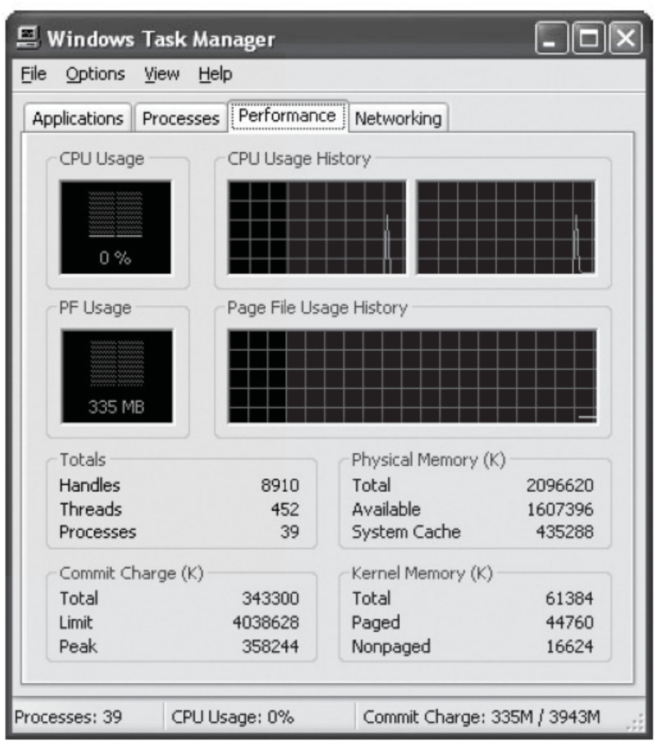


Figure 4-29 Frontside and backside buses

Multithreading At the peak of the single-CPU 32-bit computing days, Intel released a CPU called the Pentium 4 that took parallelism to the next step with Hyper-Threading. *Hyper-Threading* enabled the Pentium 4 to run multiple threads at the same time, what's generically called *simultaneous multithreading*, effectively turning the CPU into two CPUs on one chip—with a catch.

Figure 4-30 shows the Task Manager in an ancient Windows XP computer on a system running a Hyper-Threaded Pentium 4. Note how the CPU box is broken into two groups—Windows thinks this one CPU is two CPUs.

Figure 4-30 Windows Task Manager with the Performance tab displayed for a system running a Hyper-Threaded Pentium 4



Multithreading enhances a CPU's efficiency, but with a couple of limitations. First, the operating system and the application have to be designed to take advantage of the feature. Second, although the CPU simulates the actions of a second processor, it doesn't double the processing power, because the main execution resources are not duplicated.

SIM This is a great time to head over to the Chapter 4 Show! and Click! sims to see how to download and use the CPU-Z utility. Head over to totalsem .com/90x and check out the "What is CPU-Z" sim.

Multicore Processing

CPU clock speeds hit a practical limit of roughly 4 GHz around 2002–2003, motivating the CPU makers to find new ways to get more processing power for CPUs. Although Intel and AMD had different opinions about 64-bit CPUs, both decided at virtually the same time to combine two CPUs (or *cores*) into a single chip, creating a *dual-core* architecture. A dual-core CPU has two execution units—two sets of pipelines—but the two sets of pipelines share caches and RAM.

Today, multicore CPUs—with four, six, or eight cores—are common. With each generation of multicore CPU, both Intel and AMD have tinkered with the mixture of how to allocate the cache among the cores. Figure 4-31 shows another screenshot of CPU-Z, this time displaying the cache breakdown of a Haswell-based Core i7.

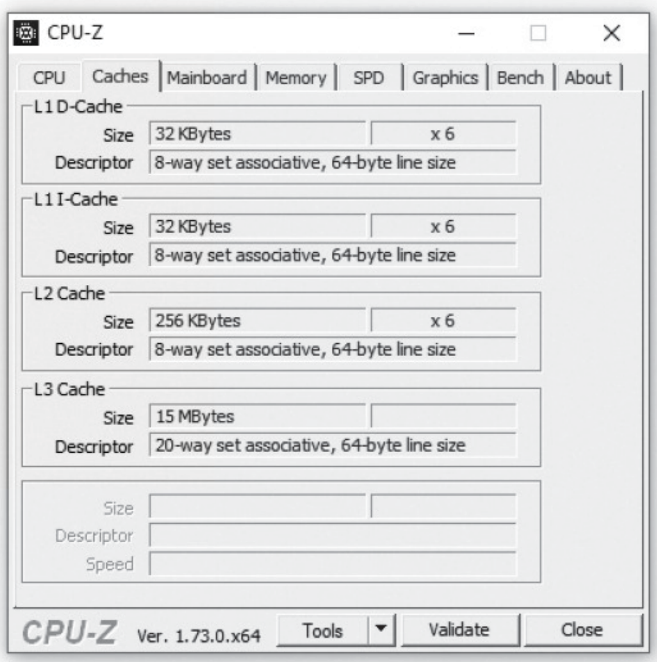


Figure 4-31 CPU-Z showing the cache details of a Haswell Core i7

Figure 4-31 reveals specific details about how this Intel CPU works with the cache. The Core i7 has L1, L2, and L3 caches of 64 KB, 256 KB, and 15 MB, respectively. (The L1 cache divides into 32 KB to handle data—the *D-Cache*—and another 32 KB for instructions—the *I-Cache*.) Each core has dedicated L1 and L2 caches. (You can tell this by the ×6 to the right of the capacity listing.) All six cores share the giant L3 cache. That pool of memory enables the cores to communicate and work together without having to access the radically slower main system RAM as much. CPU manufacturers engineered the cores in multicore CPUs to divide up work independently of the OS, known as *multicore processing*. This differs from Hyper-Threading, where the OS and applications have to be written specifically to handle the multiple threads. Note that even with multicore processors, applications have to be modified or optimized for this parallelism to have a huge impact on performance.

Integrated Memory Controller

Almost all current microprocessors have an *integrated memory controller (IMC)*, moved from the motherboard chip into the CPU to optimize the flow of information into and out from the CPU. An IMC enables faster control over things like the large L3 cache shared among multiple cores.

Just like in so many other areas of computing, manufacturers implement a variety of IMCs in their CPUs. In practice, this means that different CPUs handle different types and capacities of RAM. I'll save the details on those RAM variations for Chapter 5. For now, add "different RAM support" to your list of things to look at when making a CPU recommendation for a client.

Integrated Graphics Processing Unit

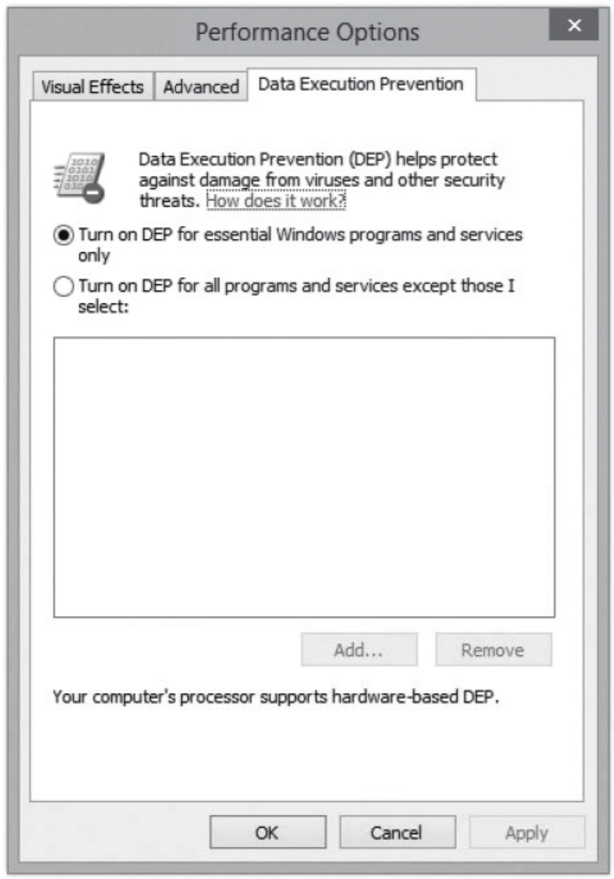
As you'll read about in much more detail in Chapter 19, the video-processing portion of the computer—made up of the parts that put a changing image on the monitor—traditionally has a discrete microprocessor that differs in both function and architecture from the CPUs designed for general-purpose computing. The generic term for the video processor is a *graphics processing unit (GPU)*. I'll spare you the details until we get to video in Chapter 19, but it turns out that graphics processors can handle certain tasks much more efficiently than the standard CPU. Integrating a GPU into the CPU enhances the overall performance of the computer while at the same time reducing energy use, size, and cost. With the proliferation of mobile devices and portable computers today, all of these benefits have obvious merit.

Both Intel and AMD produce CPUs with integrated GPUs. For many years, the quality of the GPU performance with demanding graphical programs like games made the choice between the two easy. The *Intel HD Graphics* and *Intel Iris Pro Graphics* integrated into many Core i3/i5/i7 processors pale in comparison with the AMD *accelerated processing unit* (*APU*), such as the AMD A10. AMD bought one of the two dedicated GPU manufacturers—ATI—years ago and used their technology for microprocessors with integrated CPU and GPU. (The Xbox One and PlayStation 4 gaming systems, for example, use AMD APUs.) Intel is slowly closing the gap, but isn't there as of this writing.

Security

All modern processors employ the *NX bit* technology that enables the CPU to protect certain sections of memory. This feature, coupled with implementation by the operating system, stops malicious attacks from getting to essential operating system files. Microsoft calls the feature Data Execution Prevention (DEP), turned on by default in every OS since Windows XP (see Figure 4-32).

Figure 4-32 DEP in Windows 8.1



The bad news for you is that everybody calls the NX bit technology something different:

- Intel: XD bit (eXecute Disable)
- AMD: Enhanced Virus Protection
- ARM: XN (eXecute Never)
- CompTIA: Disable execute bit

Selecting and Installing CPUs

Now that you know how CPUs work, it's time to get practical. This last section discusses selecting the proper CPU, installing several types of processors, and troubleshooting the few problems techs face with CPUs.

Selecting a CPU

When selecting a CPU, you need to make certain you get one that the motherboard can accommodate. Or, if you're buying a motherboard along with the CPU, then get the right CPU for the intended purpose. Chapter 12 discusses computer roles and helps you select the proper components for each role. You need to have a lot more knowledge of all the pieces around the CPU to get the full picture, so we'll wait until then to discuss the "why" of a particular processor. Instead, this section assumes you're placing a new CPU in an already-acquired motherboard. You need to address two key points in selecting a CPU that will work. First, does the motherboard support Intel or AMD CPUs? Second, what socket does the motherboard have?

To find answers to both those questions, you have two sources: the motherboard book or manual and the manufacturer's Web site. Figure 4-33 shows a manual for an Asus motherboard open to reveal the supported processors and the socket type.

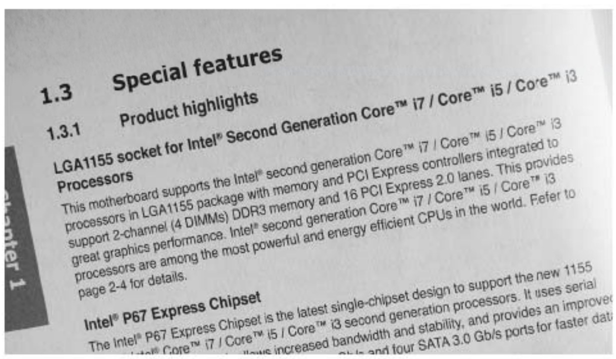


Figure 4-33 Supported processors and socket type

Just as Intel and AMD make many types of CPUs, motherboards are manufactured with various different types of sockets. The CompTIA A+ exams expect you to know which sockets go with which family of CPU. Table 4-2 charts the important Intel ones; Table 4-3 lists the AMD-based sockets. I would show you all the pictures, but, frankly, CPU sockets aren't the sexiest part of the computer.

EXAM TIP I've included the number of socket pins for the AMD-based sockets because related questions have been on the CompTIA A+ certification exams in the past. You don't need to memorize the Intel numbers, because Intel names the sockets by the number of pins.

Socket	CPU
LGA 775[1]	Pentium 4, Celeron, Pentium 4 Extreme Edition, Core 2 Duo, Core 2 Quad, Xeon, and many others
LGA 1156[2]	Core i3/i5/i7, Pentium, Celeron, Xeon
LGA 1155[3]	Core i3/i5/i7, Pentium, Celeron, Xeon
LGA 1366[4]	Core i7, Xeon, Celeron
LGA 2011[5]	Core i7, Core i7 Extreme Edition, Xeon
LGA 1150[6]	Core i3/i5/i7, Pentium, Celeron, Xeon
LGA 1151[7]	Core i3/i5/i7, Pentium, Celeron, Xeon

[1] The LGA 775 socket was the only desktop or server socket used for many years by Intel and thus just about every branded Intel CPU used it at one time or another.
[2] Socket LGA 1156 CPUs are based on the pre–Sandy Bridge architecture.
[3] Socket LGA 1155 CPUs are based on Sandy Bridge or Ivy Bridge architecture.
[4] The very first Core i7 processors used LGA 1366. Socket 1366 does not support integrated graphics.
[5] Intel uses LGA 2011 for several generations of Core i7 and Core i7 Extreme Edition CPUs. Socket 2011 does not support integrated graphics. Plus, the retail version does not come with an OEM fan and heat-sink assembly. You need to buy that separately.
[6] Socket 1150 CPUs are based on Haswell or Broadwell architecture.
[7] Socket 1151 CPUs are based on Skylake architecture. Intel had just started releasing the first of these as we went to print. Skylake will not be on the 901 exam.

Table 4-2 Intel-based Sockets

Socket	Pins	CPU
AM3[1]	941	Phenom II, Athlon II, Sempron, Opteron
AM3+	942	FX
FM1	905	A-Series[2]
FM2	904	A-Series
FM2+	906	A-Series
G34	1974	Opteron
C32	1207	Opteron

[1] The names of some of the processors designed for Socket AM3 match the names of CPUs designed for earlier sockets, but they're *not* the same CPUs. They are specific to AM3 because they support different types of RAM (see Chapter 5). Just to make things even crazier, though, AM3 CPUs work just fine in earlier Socket AM2/2+ motherboards.
[2] The A-Series features integrated GPUs and other chips.

Table 4-3 AMD-based Sockets

Deciphering Processor Numbers

Intel and AMD use different processor numbering schemes that help you compare multiple CPUs with similar names, such as Core i5. Intel's system is pretty straightforward; AMD's is muddled. Here's the scoop on both.

Intel processor numbers follow a very clear pattern. An Intel Core i7 5775 C processor, for example, maps out like this:

- Intel Core = brand
- i7 = brand modifier
- 5 = generation
- 775 = SKU numbers
- C = alpha suffix (C indicates that it's a desktop processor with integrated graphics, socket LGA 1150)

Contrast the previous processor with an Intel Core i7 5950 H Q, where the numbers map like this:

- Intel Core = brand
- i7 = brand modifier
- 5 = generation
- 950 = SKU numbers
- HQ = alpha suffix (HQ indicates that it's a mobile quad-core processor with integrated graphics)

AMD started out loving techs. Here's the breakdown for an AMD FX-8350:

- AMD = brand
- FX = product line
- 8 = series and number of processing cores
- 3 = generation (higher number means it's more refined)
- 50 = model number (higher number means it's faster)

With their A-Series, though, AMD fell a little out of love. Here's the breakdown for an AMD A10-6800K:

- AMD = brand
- A10 = product line
- 6 = generation (higher number means it's more refined, usually, though not always)
- 800 = model number (higher number means it's faster)

- K = suffix (K means that the processor has an unlocked core, designed to make overclocking easier; no suffix means it's a locked desktop core; M suffix denotes mobile version)

  Note that none of the A-Series processor numbers tell you how many cores the CPU has. Most of the desktop versions have four cores. Portable and low-powered versions (E- and C-series) have one or two cores. You have to read the packaging or search online for confirmation with any given processor.

  It's a lot to take in, especially for new techs. The good news is you won't find product numbers on the CompTIA A+ exams. The other good news is that you can refer to the this section (and the Internet) to help you choose the right processor for your customer/user. The bad news is that it's complicated.

Installation Issues

When installing a CPU, you need to use caution with the tiny pins. Plus, you must make certain that the power supply can supply enough electricity for the processor to function along with all the other components on the computer. You have to provide adequate cooling. Finally, you can decide whether to leave the CPU at stock settings or overclock it.

Socket Types

When installing a CPU, you need to exercise caution not to bend any of the tiny pins. The location of the pins differs between Intel and AMD. With Intel-based motherboards, the sockets have hundreds of tiny pins that line up with contacts on the bottom of the CPU (see Figure 4-34). Intel CPUs use a *land grid array* (*LGA*) package for socketed CPUs, where the underside of the CPU has hundreds of contact points that line up with the socket pins.

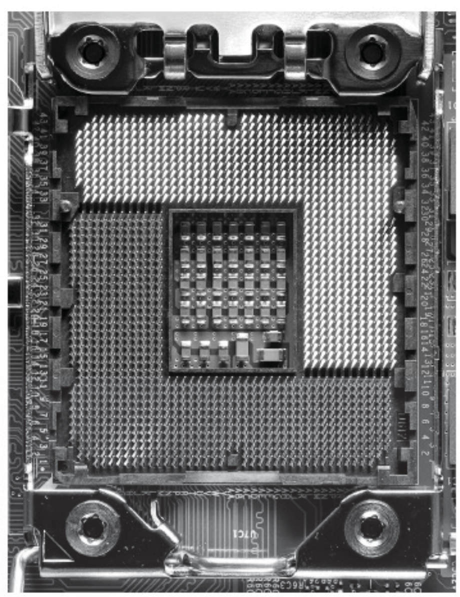


Figure 4-34 Intel-based socket with pins

AMD CPUs have the pins (see Figure 4-35); the sockets have holes. The pins on the AMD *pin grid array* (*PGA*) CPUs align with the holes in the sockets.

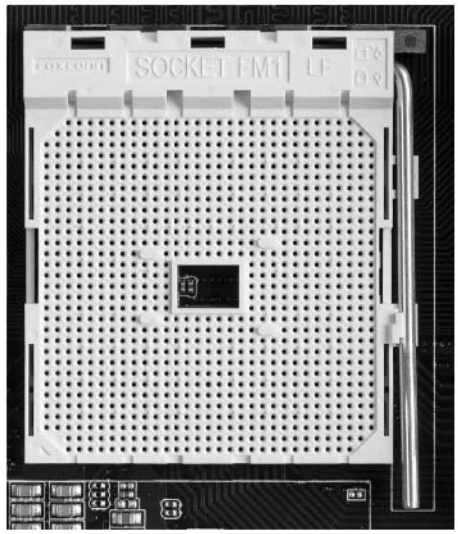


Figure 4-35
AMD-based socket without pins

All CPUs and sockets are keyed so you can't (easily) insert them incorrectly. Look at the underside of the CPU in Figure 4-36 (left). Note that the pins do not make a perfect square, because a few are missing. Now look at the top of the CPU (right). See the little mark at the corner? The socket also has tiny markings so you can line the CPU up properly with the socket.

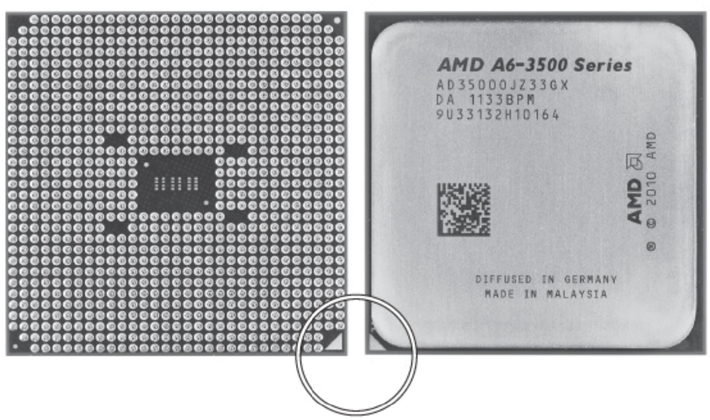


Figure 4-36
Underside and top of a CPU

In both socket styles, you release the retaining mechanism by pushing the little lever down slightly and then away from the socket (see Figure 4-37). You next raise the arm fully, and then move the retaining bracket (see Figure 4-38).

Align the processor with the socket and gently drop the processor into place. If it doesn't go in easily, check the orientation and try again. These sockets are generically called *zero insertion force (ZIF) sockets*, which means you never have to use any force at all.

Figure 4-37 Moving the release arm

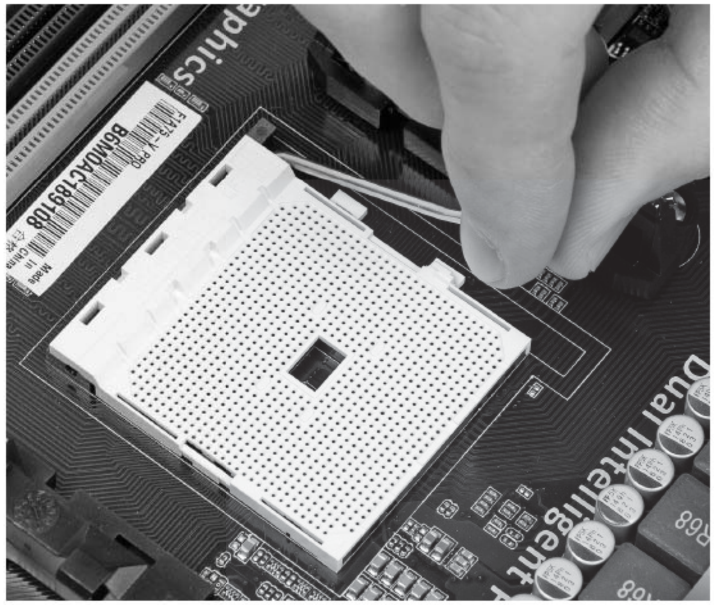
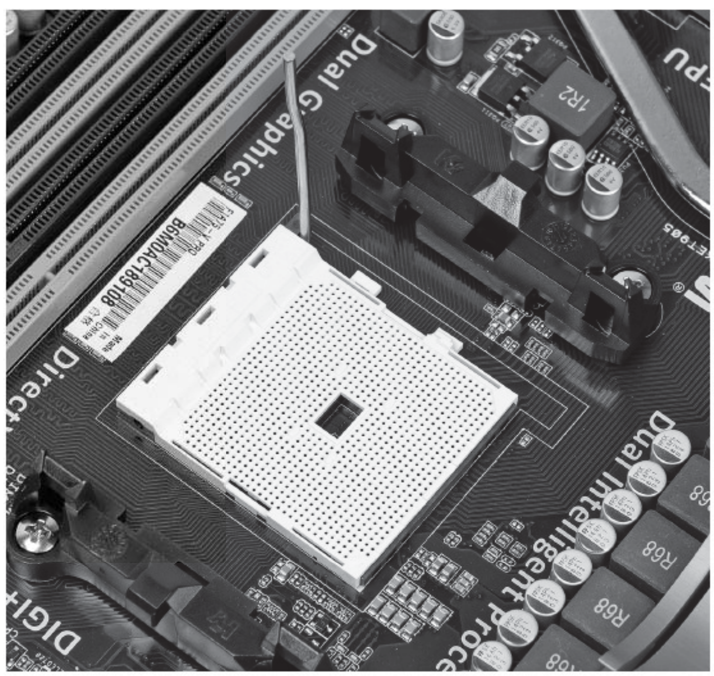


Figure 4-38 Fully opened socket



Cooling

CPUs work very hard and thus require power to function. In electrical terms, CPUs consume *wattage*, or *watts*, a unit of electrical power, just like a 100-watt light bulb consumes power whenever it's on. (See Chapter 8 for more details about electricity.) Have you ever touched a light bulb after it's been on for a while? Ouch! CPUs heat up, too.

To increase the capability of the CPUs to handle complex code, CPU manufacturers have added a lot of microscopic transistors over the years. The more transistors the CPU has, the more power they need and thus the hotter they get. CPUs don't tolerate heat well, and modern processors need active cooling solutions just to function at all. Almost every CPU uses a combination of a heat-sink and fan assembly to wick heat away from the CPU. Figure 4-39 shows the standard Intel *heat sink* and fan. Here are some cooling options:

Figure 4-39
Intel stock heat-sink and fan assembly

- **OEM CPU coolers** Original equipment manufacturer (OEM) heat-sink and fan assemblies are included with most Intel retail-boxed CPUs. OEM in this case means that Intel makes the heat-sink/fan assemblies. Rather confusingly, you'll see the term "OEM CPUs" used to mean CPUs you buy in bulk or not in the retail packaging. These are still made by Intel or AMD and are functionally identical to the retail versions. They don't come bundled with CPU coolers. Crazy, isn't it? OEM CPU coolers have one big advantage: you know absolutely they will work with your CPU. Intel Socket 2011 CPUs do not come with heat sinks or fans.
- **Specialized CPU coolers** Many companies sell third-party heat-sink and fan assemblies for a variety of CPUs. These usually exceed the OEM heat sinks in the amount of heat they dissipate. These CPU coolers invariably come with eye-catching designs to look really cool inside your system—some are even lighted (see Figure 4-40).

Figure 4-40 Cool retail heat sink

The last choice is the most impressive of all: liquid cooling! *Liquid cooling* works by running some liquid—usually water—through a metal block that sits on top of your CPU, absorbing heat. The liquid gets heated by the block, runs out of the block and into something that cools the liquid, and is then pumped through the block again. Any liquid-cooling system consists of three main parts:

- A hollow metal block that sits on the CPU
- A pump to move the liquid around
- Some device to cool the liquid

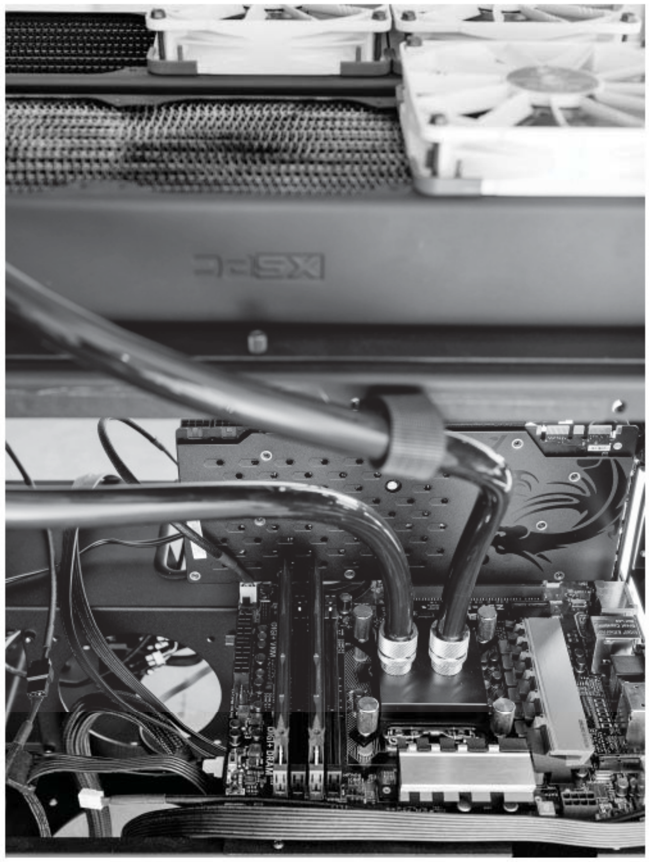
And of course, you need plenty of hosing to hook them all together. Figure 4-41 shows a typical liquid-cooled CPU.

A number of companies sell these liquid-based cooling systems. Although they look impressive and certainly cool your CPU, unless you're overclocking or want a quiet system, a good fan will more than suffice.

EXAM TIP In some instances, you can create a system that has no fan for the CPU, what's called *fanless* or *passive cooling*. Aside from mobile devices (like an Apple iPad) that have no fans, the term can be very misleading. The Xeon CPUs powering the servers in my office, for example, only have heat sinks with no fans. On the other hand, they have ducts directly to the case fans, which serve the same function as a an active CPU fan. So, go figure.

See also the "Beyond A+" section at the end of this chapter for interesting passive developments.

Figure 4-41 Liquid-cooled CPU



Once you've got a heat-sink and fan assembly sorted out, you need to connect them to the motherboard. To determine the orientation of the heat-sink and fan assembly, check the power cable from the fan. Make sure it can easily reach the three- or four-wire standout on the motherboard (see Figure 4-42). If it can't, rotate the heat sink until it can. (Check the motherboard manual if you have trouble locating the CPU fan power standout.)

Next, before inserting the heat sink, you need to add a small amount of *thermal paste* (also called *thermal compound*, *heat dope*, or *nasty silver goo*). Many heat sinks come with some thermal paste already on them; the thermal paste on these pre-doped heat sinks is covered by a small square of tape—take the tape off before you snap it to the CPU. If you need to put thermal paste on from a tube, know that you need to use only a tiny amount of this compound (see Figure 4-43). Spread it on as thinly, completely, and evenly as you can. Unlike so many other things in life, you *can* have too much thermal paste!

You secure heat sinks in various ways, depending on the manufacturer. Stock Intel heat sinks have four plungers that you simply push until they click into place in corresponding

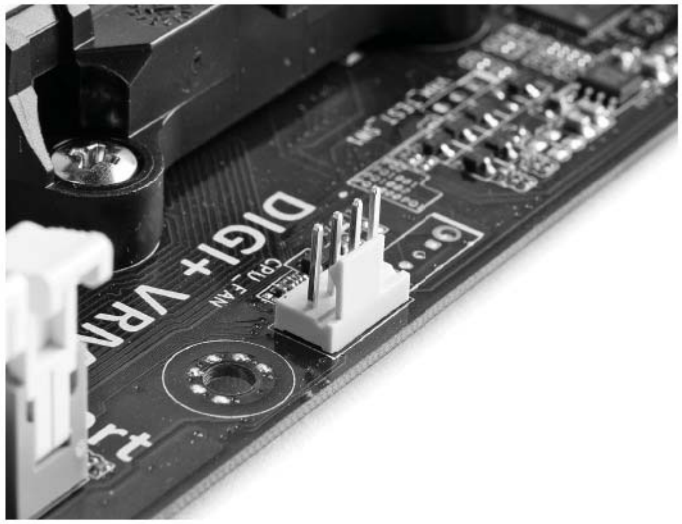


Figure 4-42 CPU fan power standout on motherboard

holes in the motherboard. AMD stock heat sinks generally have a bracket that you secure to two points on the outside of the CPU socket and a latch that you swivel to lock it down (see Figure 4-44).

Finally, you can secure many aftermarket heat-sink and fan assemblies by screwing them down from the underside of the motherboard (see Figure 4-45). You have to

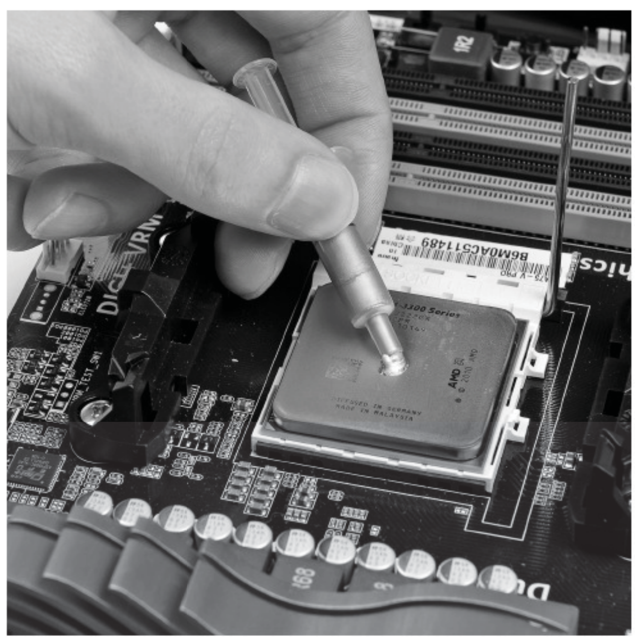


Figure 4-43 Applying thermal paste

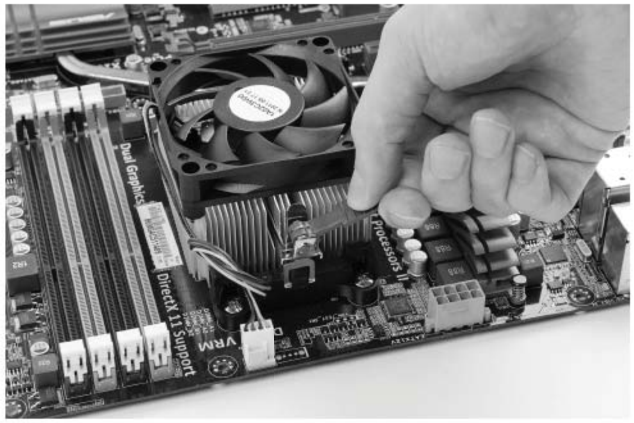


Figure 4-44
AMD stock heat-sink and fan assembly

remove the motherboard from the case or install the heat sink before you put the motherboard in the case.

For the final step, plug the fan power connector into the motherboard standout. It won't work if you don't!

Overclocking

For the CPU to work, the motherboard speed, multiplier, and voltage must be set properly. In most modern systems, the motherboard uses the CPUID functions to set these options automatically. Some motherboards enable you to adjust these settings manually by moving a jumper, changing a CMOS setting, or using software; many enthusiasts deliberately change these settings to enhance performance.

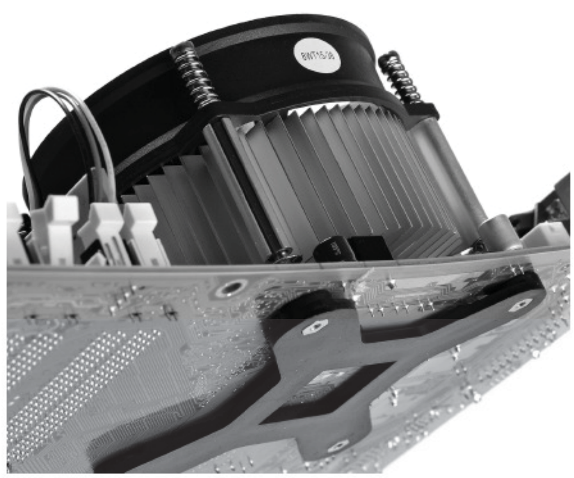


Figure 4-45
Heat-sink and fan assembly mounted to motherboard with screws

NOTE Chapter 6 goes into gory detail about the system setup utility and the area in which it stores important data (called *CMOS*), but invariably students want to experiment at this point, so I'll give you some information now. You can access the system setup utility by pressing some key as the computer starts up. This is during the text phase, well before it ever says anything about starting Windows. Most systems require you to press the DELETE key, but read the screen for details. Just be careful once you get into the system setup utility not to change anything you don't understand. And read Chapter 6!

Starting way back in the days of the Intel 80486 CPU, people intentionally ran their systems at clock speeds higher than the CPU was rated, a process called *overclocking*, and it worked. Well, *sometimes* the systems worked, and sometimes they didn't. Intel and AMD have a reason for marking a CPU at a particular clock speed—that's the highest speed they guarantee will work.

Before I say anything else, I must warn you that intentional overclocking of a CPU immediately voids most warranties. Overclocking has been known to destroy CPUs. Overclocking might make your system unstable and prone to lockups and reboots. I neither applaud nor decry the practice of overclocking. My goal here is simply to inform you of the practice. You make your own decisions.

CPU makers do not encourage overclocking. Why would you pay more for a faster processor when you can take a cheaper, slower CPU and just make it run faster? Bowing to enthusiast market pressure, both Intel and AMD make utilities that help you overclock their respective CPUs.

- **Intel Extreme Tuning Utility (Intel XTU)** Don't skip the additional Performance Tuning Protection Plan if you go this route.
- **AMD Overdrive Utility** No extra warranty is provided here; you're on your own.

Most people make a couple of adjustments to overclock successfully. First, through jumpers, CMOS settings, or software configuration, you would increase the bus speed for the system. Second, you often have to increase the voltage going into the CPU by just a little to provide stability. You do that by changing a jumper or CMOS setting (see Figure 4-46).

Overriding the defaults can completely lock up your system, to the point where even removing and reinstalling the CPU doesn't bring the motherboard back to life. (There's also a slight risk of toasting the processor, although all modern processors have circuitry that shuts them down quickly before they overheat.) Most motherboards have a jumper setting called *CMOS clear* (see Figure 4-47) that makes the CMOS go back to default settings. Before you try overclocking on a modern system, find the CMOS-clear jumper and make sure you know how to use it! Hint: Look in the motherboard manual.

To clear the CMOS, turn off the PC. Then locate one of those tiny little plastic pieces (officially called a *shunt*) and place it over the two jumper wires for a moment. Next, restart the PC and immediately go into CMOS and restore the settings you need.

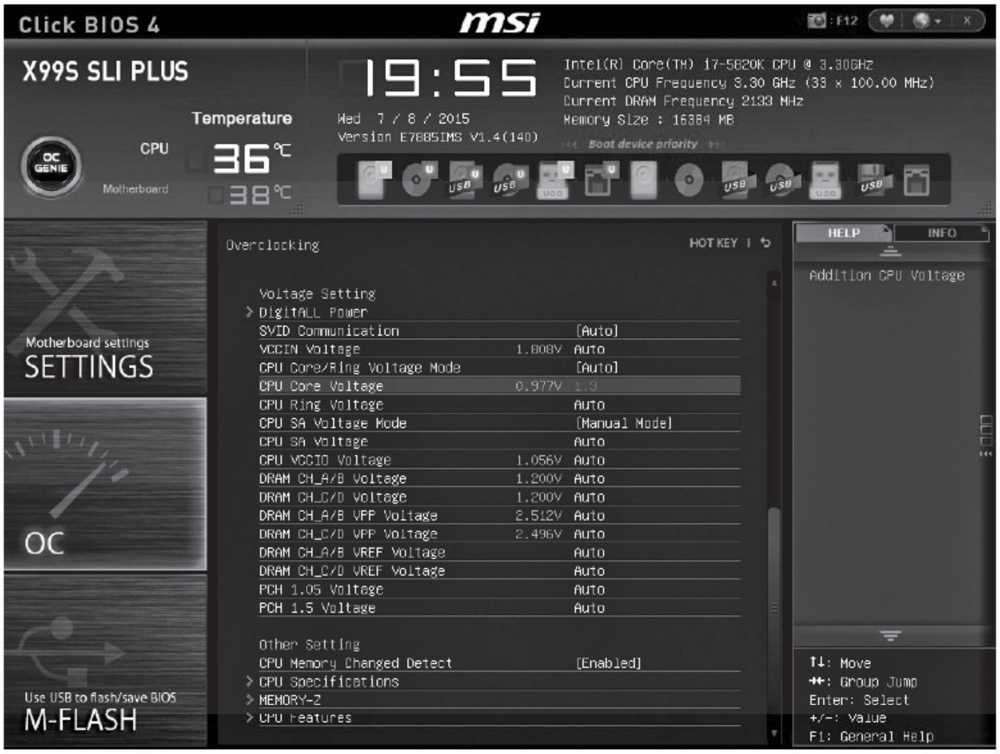
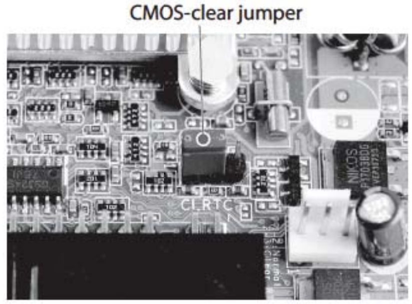


Figure 4-46 Manually overriding CPU settings in the system setup utility

Figure 4-47 CMOS-clear jumper



Troubleshooting CPUs

Troubleshooting CPU issues falls into two categories: overheating and catastrophic failures, with overheating being far more common than the latter. Once a CPU is installed properly and functioning, it rarely causes problems. The only exception is when you ask a CPU to do too much too quickly. Then you'll get a sluggish PC. The Intel Atom processor in my vintage netbook, for example, does a great job at surfing the Web, working on e-mail, and writing stellar chapters in your favorite textbook. But if you try to play a game more advanced than Half-Life (the original, circa 1998), the machine stutters and complains and refuses to play nice.

The vast majority of problems with CPUs come from faulty installation or environmental issues that cause overheating. Very rarely will you get a catastrophic failure, but we'll look at the signs of that, too.

Symptoms of Overheating

Failure to install a CPU properly results in either nothing—that is, you push the power button and nothing at all happens—or a system lock-up in a short period of time. Because of the nature of ZIF sockets, you're almost guaranteed that the issue isn't the CPU itself, but rather the installation of the heat-sink and fan assembly. Here's a checklist of possible problems that you need to address when faced with a CPU installation problem:

1. Too much thermal paste can impede the flow of heat from the CPU to the heat sink and cause the CPU to heat up rapidly. All modern CPUs have built-in fail-safes that tell them to shut down before getting damaged by heat.
2. Not enough thermal paste or thermal paste spread unevenly can cause the CPU to heat up and consequently shut itself down.
3. Failure to connect the fan power to the motherboard can cause the CPU to heat up and shut itself down.

The fan and heat-sink installation failures can be tricky the first few times you encounter them. You might see the text from the system setup. You might even get into an installation of Windows before the crash happens. The key is that as soon as you put the CPU under load—that is, make it work for a living—it heats up beyond where the faulty heat-sink connection can dissipate the heat and then shuts down.

With a system that's been running fine for a while, environmental factors can cause problems. An air conditioning failure in my office last summer, deep in the heart of very hot Texas, for example, caused machines throughout the office to run poorly. Some even shut down entirely. (At that point it was time to close the doors and send the staff to the beach, but that's another story.) A client called the other day to complain about his computer continuously rebooting and running slowly. When I arrived on the scene, I found a house with seven cats. Opening up his computer case revealed the hairy truth: the CPU fan was so clogged with cat hair that it barely spun at all! A quick cleaning with a computer vacuum and a can of compressed air and he was a happily computing client.

The CPU needs adequate ventilation. The CPU fan is essential, of course, but the inside of the case also needs to get hot air out through one or more exhaust fans and cool air in through the front vent. If the intake vent is clogged or the exhaust fans stop working or are blocked somehow, the inside of the case can heat up and overwhelm the CPU cooling devices. This will result in a system running slowly or spontaneously rebooting.

Catastrophic Failure

You'll know when a catastrophic error occurs. The PC will suddenly get a Blue Screen of Death (BSoD), what's technically called a Windows Stop error (see Figure 4-48). On Mac OS X, by comparison, you'll get a pin wheel on the screen that just doesn't go away or stop spinning. (CompTIA calls the BSoD and pin wheel *proprietary crash screens*. Most users just find them annoying.)

Or the entire computer will simply stop and go black, perhaps accompanied by a loud pop. The acrid smell of burnt electronics or ozone will grace your nasal passages. You might even see trails of smoke coming out of the case. You might not know immediately that the CPU has smoked, but follow your nose. Seriously. Sniff the inside of the case until you find the strongest smell. If it's the CPU, that's bad news. Whatever electrical short hit, it probably caused damage to the motherboard too, and you're looking at a long day of replacement and rebuilding.

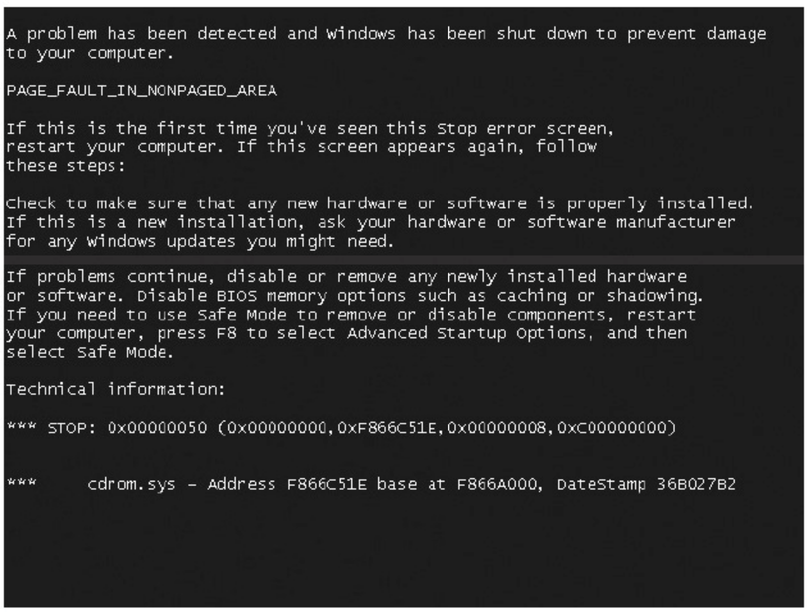


Figure 4-48 Blue Screen of Death

Beyond A+

Intel Core M

The Intel Core M runs cool and sips juice for incredibly long battery life in mobile devices. The official thermal design power (TDP) is just 4.5 watts—compared to a mobile version of a Core i7 that demands 57 watts. The trade-off Intel makes with the

Core M is in raw processing power. It falls in between the Atom and a mobile Core i3—enough to get the job done, but not enough to run a serious game or other demanding application. On the other hand, the incredibly low electricity use means manufacturers can skip the fan and make super skinny devices.

At the time of this writing, only a few portable computers run the Core M, most notably the Apple MacBook. Expect the Core M to migrate to some nonportable systems, especially Media Center PCs, where the quiet of fanless computing makes a lot of sense.

Chapter Review

Questions

1. What do registers provide for the CPU?
   - **A.** Registers determine the clock speed.
   - **B.** The CPU uses registers for temporary storage of internal commands and data.
   - **C.** Registers enable the CPU to address RAM.
   - **D.** Registers enable the CPU to control the address bus.
2. What function does the external data bus have in the PC?
   - **A.** The external data bus determines the clock speed for the CPU.
   - **B.** The CPU uses the external data bus to address RAM.
   - **C.** The external data bus provides a channel for the flow of data and commands between the CPU and RAM.
   - **D.** The CPU uses the external data bus to access registers.
3. What is the function of the address bus in the PC?
   - **A.** The address bus enables the CPU to communicate with the memory controller chip.
   - **B.** The address bus enables the memory controller chip to communicate with the RAM.
   - **C.** The address bus provides a channel for the flow of data and commands between the CPU and RAM.
   - **D.** The address bus enables the CPU to access registers.
4. Which of the following terms are measures of CPU speed?
   - **A.** Megahertz and gigahertz
   - **B.** Megabytes and gigabytes
   - **C.** Megahertz and gigabytes
   - **D.** Frontside bus, backside bus

5. Which CPU feature enables the microprocessor to support running multiple operating systems at the same time?
   - **A.** Clock multiplying
   - **B.** Caching
   - **C.** Pipelining
   - **D.** Virtualization support
6. Into which socket could you place an Intel Core i5?
   - **A.** Socket LGA 775
   - **B.** Socket LGA 1155
   - **C.** Socket C
   - **D.** Socket AM3+
7. Which feature enables a single-core CPU to function like two CPUs?
   - **A.** Hyper-Threading
   - **B.** SpeedStep
   - **C.** Virtualization
   - **D.** *x*64
8. What steps do you need to take to install a Core i3 CPU into an FM2 motherboard?
   - **A.** Lift the ZIF socket arm; place the CPU according to the orientation markings; snap on the heat-sink and fan assembly.
   - **B.** Lift the ZIF socket arm; place the CPU according to the orientation markings; add a dash of thermal paste; snap on the heat-sink and fan assembly.
   - **C.** Lift the ZIF socket arm; place the CPU according to the orientation markings; snap on the heat-sink and fan assembly; plug in the fan.
   - **D.** Take all of the steps you want to take because it's not going to work.
9. A client calls to complain that his computer starts up, but crashes when Windows starts to load. After a brief set of questions, you find out that his nephew upgraded his RAM for him over the weekend and couldn't get the computer to work right afterward. What could be the problem?
   - **A.** Thermal paste degradation
   - **B.** Disconnected CPU fan
   - **C.** Bad CPU cache
   - **D.** There's nothing wrong. It usually takes a couple of days for RAM to acclimate to the new system.

10. Darren has installed a new CPU in a client's computer, but nothing happens when he pushes the power button on the case. The LED on the motherboard is lit up, so he knows the system has power. What could the problem be?
    - **A.** He forgot to disconnect the CPU fan.
    - **B.** He forgot to apply thermal paste between the CPU and the heat-sink and fan assembly.
    - **C.** He used an AMD CPU in an Intel motherboard.
    - **D.** He used an Intel CPU in an AMD motherboard.

Answers

1. **B.** The CPU uses registers for temporary storage of internal commands and data.
2. **C.** The external data bus provides a channel for the flow of data and commands between the CPU and RAM.
3. **A.** The address bus enables the CPU to communicate with the memory controller chip.
4. **A.** The terms megahertz (MHz) and gigahertz (GHz) describe how many million or billion (respectively) cycles per second a CPU can run.
5. **D.** Intel and AMD CPUs come with virtualization support, enabling more efficient implementation of virtual machines.
6. **B.** You'll find Core i5 processors in several socket types, notably LGA 1155 and LGA 1156.
7. **A.** Intel loves its Hyper-Threading, where a single-core CPU can function like a dual-core CPU as long as it has operating system support.
8. **D.** Intel and AMD processors are not compatible at all.
9. **B.** Most likely, the nephew disconnected the CPU fan to get at the RAM slots and simply forgot to plug it back in.
10. **B.** The best answer here is that he forgot the thermal paste, though you can also make an argument for a disconnected fan.

CHAPTER 5

RAM

In this chapter, you will learn how to

- Identify the different types of DRAM packaging
- Explain the varieties of RAM
- Select and install RAM
- Perform basic RAM troubleshooting

Whenever people come up to me and start professing their computer savvy, I ask them a few questions to see how much they really know. In case you and I ever meet and you decide you want to "talk tech" with me, I'll tell you my first two questions now so you'll be ready. Both involve *random access memory* (*RAM*), the working memory for the CPU.

1. "How much RAM is in your computer?"
2. "What is RAM and why is it so important that every PC has enough?"

Can you answer either of these questions? Don't fret if you can't—you'll know how to answer both of them before you finish this chapter. Let's start by reviewing what you know about RAM thus far.

When not in use, programs and data are held in a mass storage device such as a hard disk drive (HDD), USB thumb drive, optical drive, or some other device that can hold data while the computer is off. When you load a program in Windows, your PC copies the program from the mass storage device to RAM and then runs it (see Figure 5-1).

You saw in Chapter 4 that the CPU uses *dynamic random access memory (DRAM)* as RAM for all PCs. Just like CPUs, DRAM has gone through a number of evolutionary changes over the years, resulting in improved DRAM technologies such as SDRAM, RDRAM, and DDR RAM. This chapter starts by explaining how DRAM works, and then discusses the types of DRAM used over the past several years and how they improve on the original DRAM. The third section, "Working with RAM," goes into the details of finding and installing RAM. The chapter finishes with troubleshooting RAM problems.

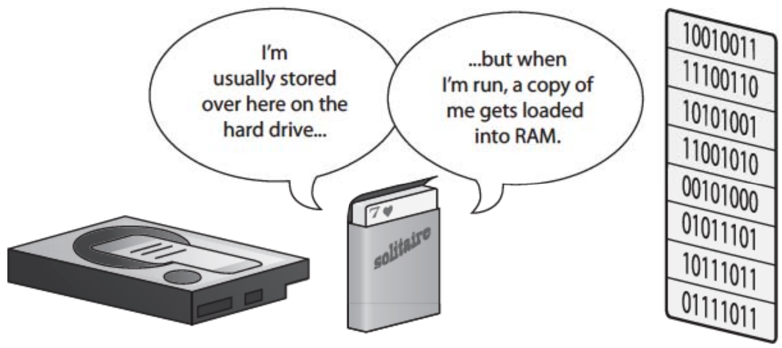


Figure 5-1 Mass storage holds programs, but programs need to run from RAM.

Historical/Conceptual

Understanding DRAM

As discussed in Chapter 4, DRAM functions like an electronic spreadsheet, with numbered rows containing cells and each cell holding a one or a zero. Now let's look at what's physically happening. Each spreadsheet cell is a special type of semiconductor that can hold a single bit—one or zero—by using microscopic capacitors and transistors. DRAM makers put these semiconductors into chips that can hold a certain number of bits. The bits inside the chips are organized in a rectangular fashion, using rows and columns.

Each chip has a limit on the number of lines of code it can contain. Think of each line of code as one of the rows on the electronic spreadsheet; one chip might be able to store a million rows of code while another chip might be able to store over a billion lines. Each chip also has a limit on the width of the lines of code it can handle. One chip might handle 8-bit-wide data while another might handle 16-bit-wide data. Techs describe chips by bits rather than bytes, so they refer to *×8* and *×16*, respectively. Just as you could describe a spreadsheet by the number of rows and columns—John's accounting spreadsheet is huge, 48 rows × 12 columns—memory makers describe RAM chips the same way. An individual DRAM chip that holds 1,048,576 rows and 8 columns, for example, would be a *1M×8* chip, with "M" as shorthand for "mega," just like in megabytes (2^{20} bytes). It is difficult if not impossible to tell the size of a DRAM chip just by looking at it—only the DRAM makers know the meaning of the tiny numbers on the chips (see Figure 5-2), although sometimes you can make a good guess.

Organizing DRAM

Because of its low cost, high speed, and capability to contain a lot of data in a relatively small package, DRAM has been the standard RAM used in all computers—not just

Figure 5-2 What do these numbers mean?

PCs—since the mid-1970s. DRAM can be found in just about everything, from automobiles to automatic bread makers.

The PC has very specific requirements for DRAM. The original 8088 processor had an 8-bit frontside bus. Commands given to an 8088 processor were in discrete 8-bit chunks. You needed RAM that could store data in 8-bit (1-byte) chunks, so that each time the CPU asked for a line of code, the memory controller chip (MCC) could put an 8-bit chunk on the data bus. This optimized the flow of data into (and out from) the CPU. Although today's DRAM chips may have widths greater than 1 bit, all DRAM chips back then were 1 bit wide, meaning only sizes such as 64 K × 1 or 256 K × 1 existed—always 1 bit wide. So how was 1-bit-wide DRAM turned into 8-bit-wide memory? The solution was quite simple: just take eight 1-bit-wide chips and use the MCC to organize them electronically to be eight wide (see Figure 5-3).

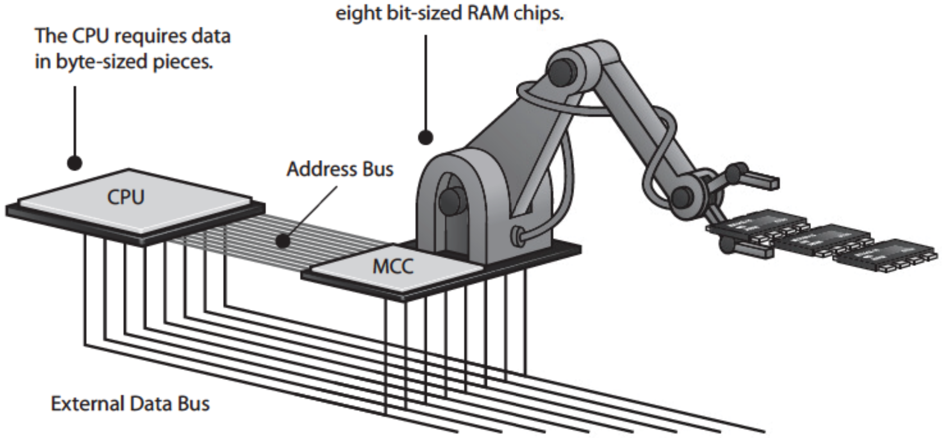


Figure 5-3 The MCC accessing data on RAM soldered onto the motherboard

Practical DRAM

Okay, before you learn more about DRAM, I need to clarify a critical point. When you first saw the 8088's machine language in Chapter 4, all the examples in the "codebook" were exactly 1-byte commands. Figure 5-4 shows the codebook again—see how all the commands are 1 byte?

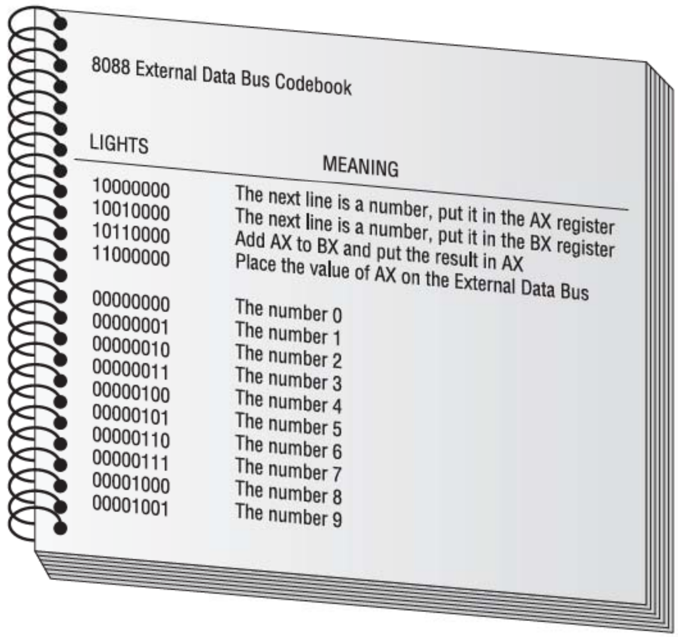


Figure 5-4 Codebook again

Well, the reality is slightly different. Most of the 8088 machine language commands are 1 byte, but more-complex commands need 2 bytes. For example, the following command tells the CPU to move 163 bytes "up the RAM spreadsheet" and run whatever command is there. Cool, eh?

```
1110100110100011
```

The problem here is that the command is 2 bytes wide, not 1 byte. So how did the 8088 handle this? Simple—it just took the command 1 byte at a time. It took twice as long to handle the command because the MCC had to go to RAM twice, but it worked.

So if some of the commands are more than 1 byte wide, why didn't Intel make the 8088 with a 16-bit frontside bus? Wouldn't that have been better? Well, Intel did. Intel invented a CPU called the 8086. The 8086 actually predates the 8088 and was absolutely identical to the 8088 except for one small detail: it had a 16-bit frontside bus. IBM could have used the 8086 instead of the 8088 and used 2-byte-wide RAM instead of 1-byte-wide RAM. Of course, they would have needed to invent an MCC that could handle that kind of RAM (see Figure 5-5).

Why did Intel sell the 8088 to IBM instead of the 8086? There were two reasons. Nobody had invented an affordable MCC or RAM that handled 2 bytes at a time. Sure, chips had been invented, but they were *expensive* and IBM didn't think anyone would want to pay $12,000 for a personal computer. So IBM bought the Intel 8088, not the Intel 8086, and all our RAM came in bytes. But as you might imagine, it didn't stay that way for long.

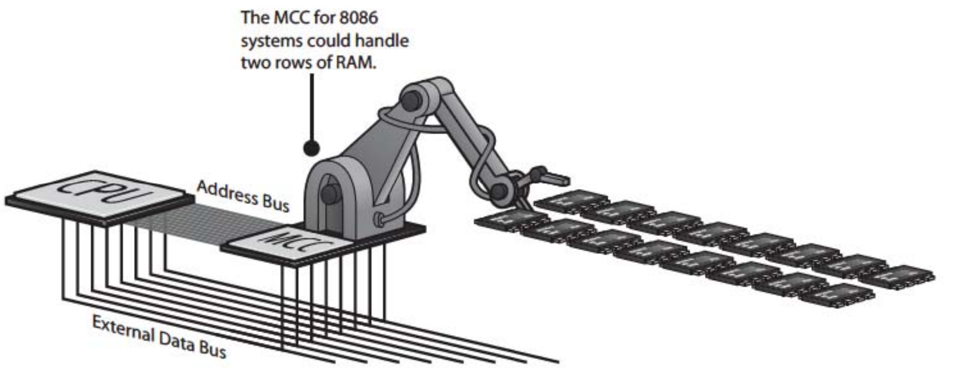


Figure 5-5 Pumped-up 8086 MCC at work

DRAM Sticks

As CPU data bus sizes increased, so too did the need for RAM wide enough to fill the bus. The Intel 80386 CPU, for example, had a 32-bit data bus and thus the need for 32-bit-wide DRAM. Imagine having to line up 32 one-bit-wide DRAM chips on a motherboard. Talk about a waste of space! Figure 5-6 shows motherboard RAM run amuck.

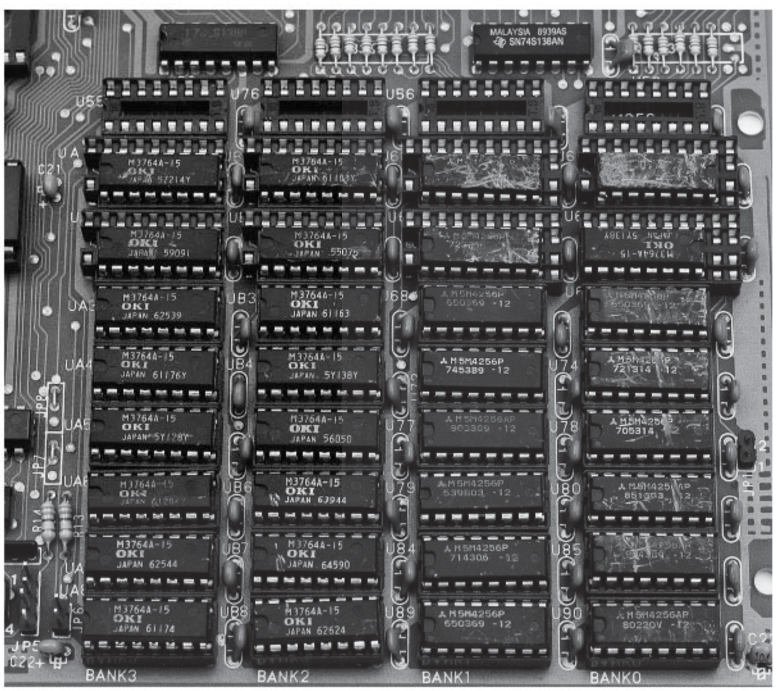
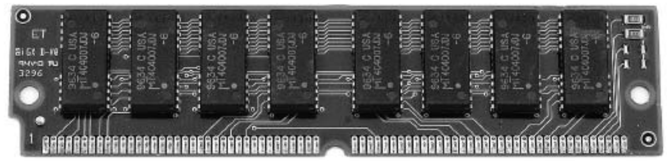


Figure 5-6 That's a lot of real estate used by RAM chips!

Figure 5-7
A 72-pin SIMM



DRAM manufacturers responded by creating wider DRAM chips, such as ×4, ×8, and ×16, and putting multiples of them on a small circuit board called a *stick* or *module*. Figure 5-7 shows an early stick, called a *single inline memory module (SIMM)*, with eight DRAM chips. To add RAM to a modern machine, you need to get the right stick or sticks for the particular motherboard. Your motherboard manual tells you precisely what sort of module you need and how much RAM you can install.

Modern CPUs are a lot smarter than the old Intel 8088. Their machine languages have some commands that are up to 64 bits (8 bytes) wide. They also have at least a 64-bit frontside bus that can handle more than just 8 bits. They don't want RAM to give them a puny 8 bits at a time! To optimize the flow of data into and out of the CPU, the modern MCC provides at least 64 bits of data every time the CPU requests information from RAM.

Try This!

Dealing with Old RAM

Often in the PC world, old technology and ways of doing things are reimplemented with some newer technology. A tech who knows these ancient ways will have extra opportunities. Many thousands of companies—including hospitals, auto repair places, and more—use very old proprietary applications that keep track of medical records, inventory, and so on. If you're called to work on one of these ancient systems, you need to know how to work with old parts, so try this.

Obtain an old computer. Ask your uncle, cousin, or Great Aunt Edna if they have a PC collecting dust in a closet that you can use. Failing that, go to a secondhand store or market and buy one for a few dollars.

Open up the system and check out the RAM. Remove the RAM from the motherboard and then replace it to familiarize yourself with the internals. You never know when some critical system will go down and need repair immediately—and you're the one to do it!

Modern DRAM sticks come in 32-bit- and 64-bit-wide data form factors with a varying number of chips. Many techs describe these memory modules by their width, so we call them ×*32* and ×*64*. Note that this number does *not* describe the width of the individual DRAM chips on the module. When you read or hear about *by whatever* memory, you need to know whether that person is talking about the DRAM width or the module

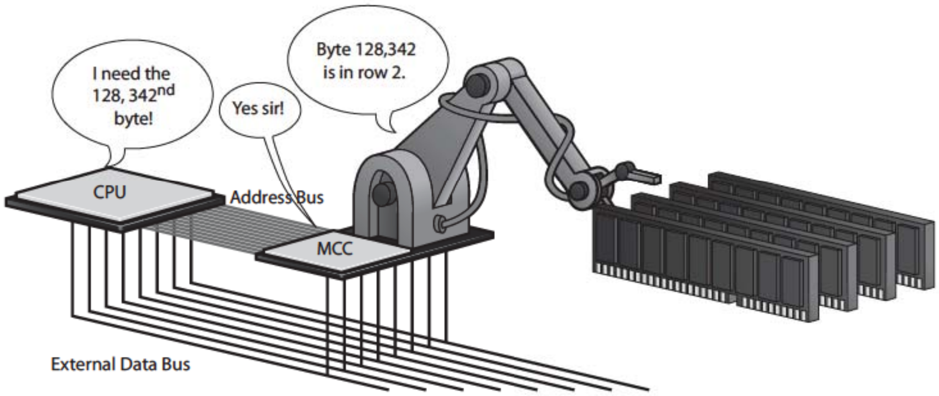


Figure 5-8 The MCC knows the real location of the DRAM.

width. When the CPU needs certain bytes of data, it requests those bytes via the address bus. The CPU does not know the physical location of the RAM that stores that data, nor the physical makeup of the RAM—such as how many DRAM chips work together to provide the 64-bit-wide memory rows. The MCC keeps track of this and just gives the CPU whichever bytes it requests (see Figure 5-8).

Consumer RAM

If modern DRAM modules come in sizes much wider than a byte, why do people still use the word "byte" to describe how much DRAM they have? Convention. Habit. Rather than using a label that describes the electronic structure of RAM, common usage describes the *total capacity of RAM on a stick in bytes*. John has a single 4-GB stick of RAM on his motherboard, for example, and Sally has two 2-GB sticks. Both systems have a total of 4 GB of system RAM. That's what your clients care about. Having enough RAM makes their systems snappy and stable; not enough RAM means their systems run poorly. As a tech, you need to know more, of course, to pick the right RAM for many different types of computers.

Types of RAM

Development of newer, wider, and faster CPUs and MCCs motivate DRAM manufacturers to invent new DRAM technologies that deliver enough data at a single pop to optimize the flow of data into and out of the CPU.

SDRAM

Most modern systems use some form of *synchronous DRAM (SDRAM)*. SDRAM is still DRAM, but it is *synchronous*—tied to the system clock, just like the CPU and MCC, so the MCC knows when data is ready to be grabbed from SDRAM. This results in little wasted time.

Figure 5-9
144-pin micro-DIMM (photo courtesy of Micron Technology, Inc.)

SDRAM made its debut in 1996 on a stick called a *dual inline memory module (DIMM)*. The early SDRAM DIMMs came in a wide variety of pin sizes. The most common pin sizes found on desktops were the 168-pin variety. Laptop DIMMs came in 68-pin, 144-pin (see Figure 5-9), or 172-pin *micro-DIMM* packages; and the 72-pin, 144-pin, or 200-pin *small-outline DIMM (SO-DIMM)* form factors (see Figure 5-10). With the exception of the 32-bit 72-pin SO-DIMM, all these DIMM varieties delivered 64-bit-wide data to match the 64-bit data bus of every CPU since the original Pentium.

To take advantage of SDRAM, you needed a PC designed to use SDRAM. If you had a system with slots for 168-pin DIMMs, for example, your system used SDRAM. A DIMM in any one of the DIMM slots could fill the 64-bit bus, so each slot was called a *bank*. You could install one, two, or more sticks and the system would work. Note that on laptops that used the 72-pin SO-DIMM, you needed to install two sticks of RAM to make a full bank, because each stick only provided half the bus width.

SDRAM was tied to the system clock, so its clock speed matched the frontside bus. Five clock speeds were commonly used on the early SDRAM systems: 66, 75, 83, 100, and 133 MHz. The RAM speed had to match or exceed the system speed or the computer would be unstable or wouldn't work at all. These speeds were prefixed with a "PC" in the front, based on a standard forwarded by Intel, so SDRAM speeds were PC66 through PC133. For a Pentium III computer with a 100-MHz frontside bus, you needed to buy SDRAM DIMMs rated to handle it, such as PC100 or PC133.

Figure 5-10
A 168-pin DIMM above a 144-pin SO-DIMM

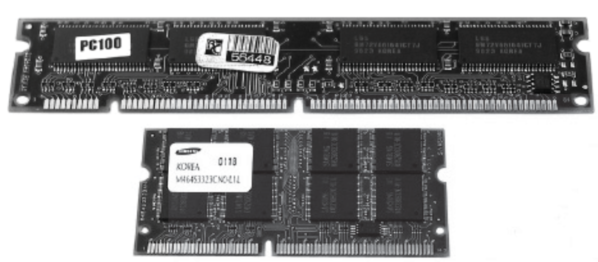


Figure 5-11
RDRAM

RDRAM

When Intel was developing the Pentium 4, they knew that regular SDRAM just wasn't going to be fast enough to handle the quad-pumped 400-MHz frontside bus. Intel announced plans to replace SDRAM with a very fast, new type of RAM developed by Rambus, Inc., called *Rambus DRAM*, or simply *RDRAM* (see Figure 5-11). Hailed by Intel as the next great leap in DRAM technology, RDRAM could handle speeds up to 800 MHz, which gave Intel plenty of room to improve the Pentium 4.

RDRAM was greatly anticipated by the industry for years, but industry support for RDRAM proved less than enthusiastic due to significant delays in development and a price many times that of SDRAM. Despite this grudging support, almost all major PC makers sold systems that used RDRAM—for a while. From a tech's standpoint, RDRAM shared almost all of the characteristics of SDRAM. A stick of RDRAM was called a *RIMM*. In this case, however, the letters didn't actually stand for anything; they just rhymed: SIMMs, DIMMs, and RIMMs, get it?

NOTE The 400-MHz frontside bus speed wasn't achieved by making the system clock faster—it was done by making CPUs and MCCs capable of sending 64 bits of data two or four times for every clock cycle, effectively doubling or quadrupling the system bus speed.

901

DDR SDRAM

AMD and many major system and memory makers threw their support behind an alternative to RDRAM, *double data rate SDRAM (DDR SDRAM)*. DDR SDRAM basically copied Rambus, doubling the throughput of SDRAM by making two processes for every clock cycle. This synchronized (pardon the pun) nicely with the Athlon and later AMD processors' double-pumped frontside bus. DDR SDRAM could not run as fast as RDRAM—although relatively low frontside bus speeds made that a moot point—but cost only slightly more than regular SDRAM.

DDR SDRAM for desktops comes in 184-pin DIMMs. These DIMMs match 168-pin DIMMs in physical size but not in pin compatibility (see Figure 5-12). The slots for the two types of RAM appear similar as well but have different guide notches, so you

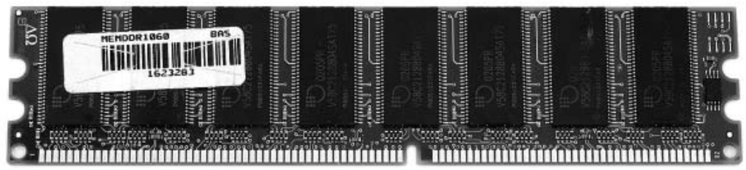


Figure 5-12
DDR SDRAM

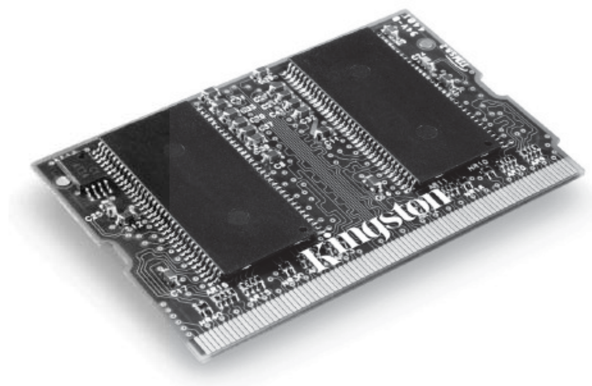


Figure 5-13
172-pin DDR SDRAM micro-DIMM (photo courtesy of Kingston/Joint Harvest)

can't insert either type of RAM into the other's slot. DDR SDRAM for laptops comes in either 200-pin SO-DIMMs or 172-pin micro-DIMMs (see Figure 5-13).

NOTE Most techs drop some or all of the SDRAM part of DDR SDRAM when engaged in normal geekspeak. You'll hear the memory referred to as DDR, DDR RAM, and the weird hybrid, DDRAM.

DDR sticks use a rather interesting naming convention based on the number of bytes per second of data throughput the RAM can handle. To determine the bytes per second, take the MHz speed and multiply by 8 bytes (the width of all DDR SDRAM sticks). So 400 MHz multiplied by 8 is 3200 megabytes per second (MBps). Put the abbreviation "PC" in the front to make the new term: PC3200. Many techs also use the naming convention used for the individual DDR chips; for example, *DDR400* refers to a 400-MHz DDR SDRAM chip running on a 200-MHz clock.

Even though the term DDR*xxx* is really just for individual DDR chips and the term PC*xxxx* is for DDR sticks, this tradition of two names for every speed of RAM is a bit of a challenge because you'll often hear both terms used interchangeably. Table 5-1 shows all the speeds for DDR—not all of these are commonly used.

Clock Speed	DDR Speed Rating	PC Speed Rating
100 MHz	DDR-200	PC-1600
133 MHz	DDR-266	PC-2100
166 MHz	DDR-333	PC-2700
200 MHz	DDR-400	PC-3200
217 MHz	DDR-433	PC-3500
233 MHz	DDR-466	PC-3700
250 MHz	DDR-500	PC-4000
275 MHz	DDR-550	PC-4400
300 MHz	DDR-600	PC-4800

Table 5-1 DDR Speeds

Following the lead of AMD and other manufacturers, the PC industry adopted DDR SDRAM as the standard system RAM. In the summer of 2003, Intel relented and stopped producing motherboards and memory controllers that required RDRAM.

One thing is certain about PC technologies: any good idea that can be copied will be copied. One of Rambus' best concepts was the *dual-channel architecture*—using two sticks of RDRAM together to increase throughput. Manufacturers have released motherboards with MCCs that support dual-channel architecture using DDR SDRAM. Dual-channel DDR motherboards use regular DDR sticks, although manufacturers often sell RAM in matched pairs, branding them as dual-channel RAM.

SIM I've got a great Chapter 5 Challenge! sim on calculating RAM speeds at totalsem.com/90x. Check it out right now!

Dual-channel DDR requires two identical sticks of DDR and they must snap into two paired slots. Many motherboards offer four slots (see Figure 5-14).

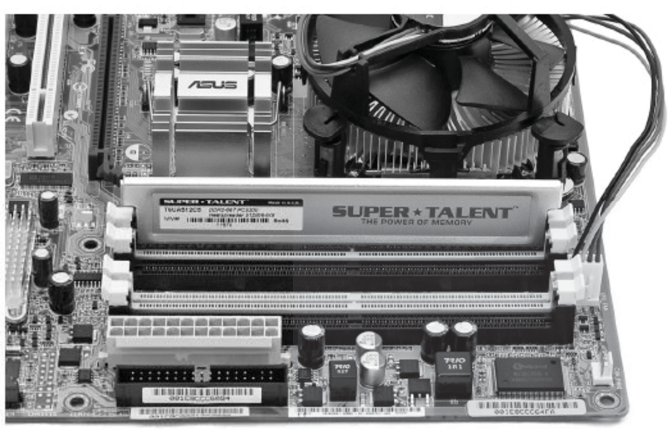


Figure 5-14 A motherboard showing four RAM slots. By populating the same-colored slots with identical RAM, you can run in dual-channel mode.

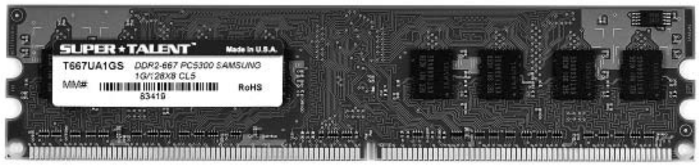


Figure 5-15 240-pin DDR2 DIMM

DDR2

DDR2 is DDR RAM with some improvements in its electrical characteristics, enabling it to run even faster than DDR while using less power. The big speed increase from DDR2 comes by clock doubling the input/output circuits on the chips. This does not speed up the core RAM—the part that holds the data—but speeding up the input/output and adding special buffers (sort of like a cache) makes DDR2 run much faster than regular DDR. DDR2 uses a 240-pin DIMM that's not compatible with DDR (see Figure 5-15). Likewise, the DDR2 200-pin SO-DIMM is incompatible with the DDR SO-DIMM. You'll find motherboards running both single-channel and dual-channel DDR2.

NOTE DDR2 RAM sticks will not fit into DDR sockets, nor are they electronically compatible.

Table 5-2 shows some of the common DDR2 speeds.

Core RAM Clock Speed	DDR I/O Speed	DDR2 Speed Rating	PC Speed Rating
100 MHz	200 MHz	DDR2-400	PC2-3200
133 MHz	266 MHz	DDR2-533	PC2-4200
166 MHz	333 MHz	DDR2-667	PC2-5300
200 MHz	400 MHz	DDR2-800	PC2-6400
266 MHz	533 MHz	DDR2-1066	PC2-8500

Table 5-2 DDR2 Speeds

DDR3

DDR3 boasts higher speeds, more efficient architecture, and around 30 percent lower power consumption than DDR2 RAM, making it a compelling choice for system builders. Just like its predecessor, DDR3 uses a 240-pin DIMM, albeit one that is slotted differently to make it difficult for users to install the wrong RAM in their system without using a hammer (see Figure 5-16). DDR3 SO-DIMMs for portable computers have 204 pins. Neither fits into a DDR2 socket.

EXAM TIP The 220-901 exam will test your knowledge of the various RAM types including DDR, DDR2, and DDR3. Be sure you are familiar with their individual characteristics and differences. DDR3 DIMMs have 240 pins, for example, and DDR3 SO-DIMMs have 204 pins. They are physically and electronically incompatible with DDR2 DIMMs and SO-DIMMs.

Figure 5-16 DDR2 DIMM on top of a DDR3 DIMM

NOTE Do not confuse DDR3 with GDDR3; the latter is a type of memory used solely in video cards. See Chapter 19, "Display Technologies," for the scoop on video-specific types of memory.

DDR3 doubles the buffer of DDR2 from 4 bits to 8 bits, giving it a huge boost in bandwidth over older RAM. Not only that, but some DDR3 modules also include a feature called *XMP*, or *extended memory profile*, that enables power users to overclock their RAM easily, boosting their already fast memory. DDR3 modules also use higher-density memory chips, up to 16-GB DDR3 modules.

Some motherboards that support DDR3 also support features called *triple-channel architecture* or *quad-channel architecture*, which work a lot like dual-channel, but with three or four sticks of RAM instead of two. Intel's LGA 1366 platform supports triple-channel memory; no AMD processors support a triple-channel feature. More recent Intel and AMD systems support quad-channel memory.

EXAM TIP Be sure you are familiar with single-, dual-, and triple-channel memory architectures.

Table 5-3 shows common DDR3 speeds. Note how DDR3 I/O speeds are quadruple the clock speeds, whereas DDR2 I/O speeds are only double the clock. This speed increase is due to the increased buffer size, which enables DDR3 to grab twice as much data every clock cycle as DDR2 can.

Core RAM Clock Speed	DDR I/O Speed	DDR3 Speed Rating	PC Speed Rating
100 MHz	400 MHz	DDR3-800	PC3-6400
133 MHz	533 MHz	DDR3-1066	PC3-8500
166 MHz	667 MHz	DDR3-1333	PC3-10667
200 MHz	800 MHz	DDR3-1600	PC3-12800
233 MHz	933 MHz	DDR3-1866	PC3-14900
266 MHz	1066 MHz	DDR3-2133	PC3-17000
300 MHz	1200 MHz	DDR3-2400	PC3-19200

Table 5-3 DDR3 Speeds

DDR3L/DDR3U

Memory manufacturers offer a low-voltage version of DDR3, most commonly labeled *DDR3L*, that provides substantial cost savings when used in massive RAM applications. (Think big data centers, like the ones that power Google.) DDR3L runs at 1.35 volts (V), compared to the 1.5 V or 1.65 V of regular DDR3, providing cost savings up to 15 percent—that adds up fast! The ultra-low-voltage version of DDR3, *DDR3U*, runs at a miserly 1.25 V.

Lower voltage means less heat generated. In a server farm or data center, that can reduce the air conditioning bill by a lot. That's a good thing.

The DIMM is slot-compatible with DDR3, although not necessarily a drama-free replacement on older motherboards. A motherboard pushing 1.5 V to the RAM slots and RAM only capable of running at 1.35 V would not result in happiness.

For best results, check the manual that came with the motherboard in question or check the manufacturer's Web site for support. Also, many RAM manufacturers produce RAM modules capable of running at 1.35 V or 1.5 V; those will work in any motherboard that supports DDR3. Some modules can handle the full gamut, from 1.25 V to 1.65 V.

DDR4

DDR4 arrived on the scene in late 2014 with much fanfare and slow adoption. DDR4 offers higher density and lower voltages than DDR3, and can handle faster data transfer rates. In theory, manufacturers could create DDR4 DIMMs up to 512 GB. As of this writing, DIMMS running DDR4 top out at 16 GB, like DDR3, but run at only 1.2 V. (There's a performance version that runs at 1.35 V and a low-voltage version at 1.05 V too.)

DDR4 uses a 288-pin DIMM, so they are not backwardly compatible with DDR3 slots. DDR4 SO-DIMMs have 260 pins that are not compatible with DDR3 204-pin SO-DIMM slots. Some motherboard manufacturers have released boards that offer support for both DDR3 and DDR4, by providing both slot types.

NOTE Intel has proposed a memory standard as of this writing that would bridge the gap between DDR3 and DDR4, called *UniDIMM*. The processor architecture released in the summer of 2015, called Skylake, can handle either type of memory. The UniDIMM memory can be either DDR3 or DDR4 and the processor can handle it.

Depending on when you're reading this note, UniDIMM might be a reality, or a blip in the past. Either way, it's not covered on the CompTIA A+ 901 exam.

With DDR4, most techs have switched from bit rate to megatransfers per second (MT/s), a way to describe the bandwidth as the number of data transfer operations happening at any given second. For DDR4, the number is pretty huge. Table 5-4 shows common DDR4 speeds and labels.

Clock Speed	Bandwidth	DDR4 Speed Rating	PC Speed Rating
200 MHz	1600 MT/s	DDR4-1600	PC4-12800
266 MHz	2133 MT/s	DDR4-2133	PC4-17000
300 MHz	2400 MT/s	DDR4-2400	PC4-19200
400 MHz	3200 MT/s	DDR4-3200	PC4-25600

Table 5-4 Standard DDR4 Varieties

EXAM TIP The CompTIA A+ 901 exam covers only DDR, DDR2, and DDR3 sticks. You won't find DDR3L, DDR3U, or DDR4 on the exam.

RAM Variations

Within each class of RAM, you'll find variations in packaging, speed, quality, and the capability to handle data with more or fewer errors. Higher-end systems often need higher-end RAM, so knowing these variations is of crucial importance to techs.

Double-Sided DIMMs

Every type of RAM stick comes in one of two types: *single-sided RAM* and *double-sided RAM*. As their name implies, single-sided sticks have chips on only one side of the stick. Double-sided sticks have chips on both sides (see Figure 5-17). Double-sided sticks are basically two sticks of RAM soldered onto one board. There's nothing wrong with double-sided RAM sticks other than the fact that some motherboards either can't use them or can only use them in certain ways—for example, only if you use a single stick and it goes into a certain slot.

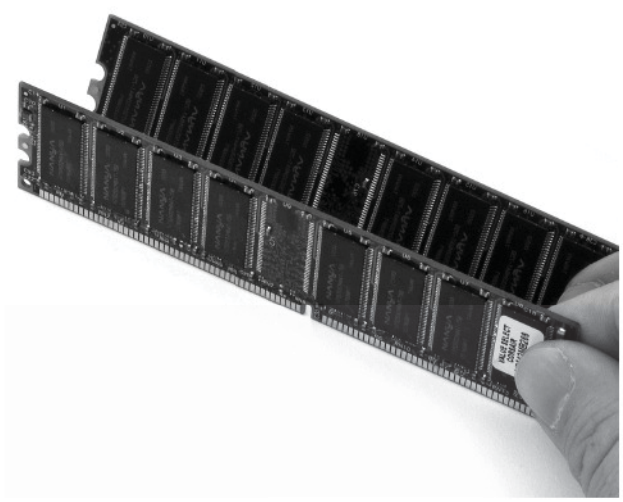


Figure 5-17 Double-sided DDR SDRAM

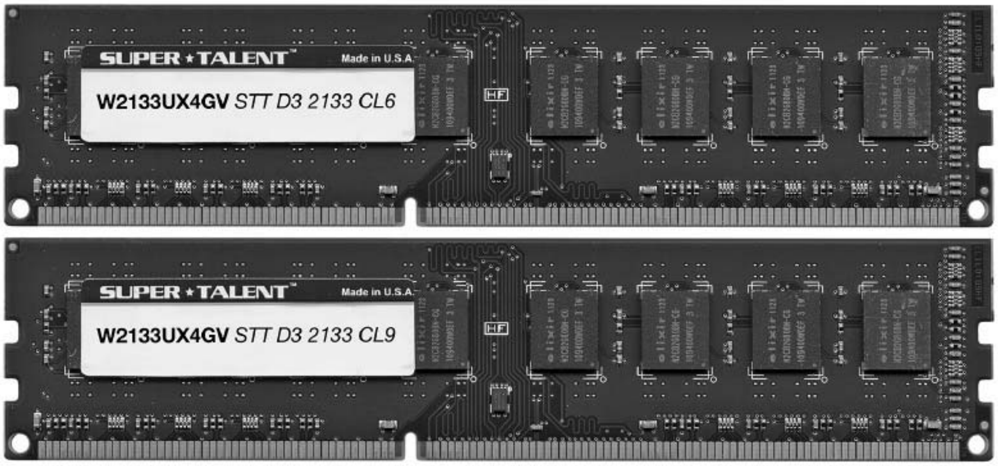


Figure 5-18 Why is one more expensive than the other?

Latency

If you've shopped for RAM lately, you may have noticed terms such as "CL6" or "low latency" as you tried to determine which RAM to purchase. You might find two otherwise identical RAM sticks with a 20 percent price difference and a salesperson pressuring you to buy the more expensive one because it's "faster" even though both sticks say DDR3-2133 (see Figure 5-18).

RAM responds to electrical signals at varying rates. When the memory controller starts to grab a line of memory, for example, a slight delay occurs; think of it as the RAM getting off the couch. After the RAM sends out the requested line of memory, there's another slight delay before the memory controller can ask for another line—the RAM sat back down. The delay in RAM's response time is called its *latency*. RAM with a lower latency—such as CL6—is faster than RAM with a higher latency—such as CL9—because it responds more quickly. The CL refers to clock cycle delays. The "6" means that the memory delays six clock cycles before delivering the requested data; the "9" means a nine-cycle delay.

The latency numbers can help you decide between two similar sticks of RAM, but can deceive you when comparing generations of memory. It's obvious that an 8-GB stick from G.SKILL with a column array strobe (CAS) latency of 11 is not as good as their "identical" and slightly more expensive stick with a CAS latency of 9. That's comparing apples to apples.

Before DDR4 debuted, however, its relatively high latency made enthusiasts question the memory companies, complaining that it would be too slow. Once released and tested, though, the added efficiency and technology improvements proved DDR4 performed equally with DDR3 at the same clock speed, even with higher latency. Plus DDR4 will eventually scale up much higher in speed that DDR3 can.

NOTE Latency numbers reflect how many ticks of the system clock it takes before the RAM responds. If you speed up the system clock—say, from 200 MHz to 266 MHz—the same stick of RAM might take an extra tick before it can respond. When you take RAM out of an older system and put it into a newer one, you might get a seemingly dead PC, even though the RAM fits in the DIMM slot. Many motherboards enable you to adjust the RAM timings manually. If yours does so, try raising the latency to give the slower RAM time to respond. See Chapter 7, "BIOS," to learn how to make these adjustments (and how to recover if you make a mistake).

From a tech's standpoint, you need to get the proper RAM for the system you're working on. If you put a high-latency stick in a motherboard set up for a low-latency stick, you'll get an unstable or completely dead PC. Check the motherboard manual or RAM manufacturer's Web site and get the quickest RAM the motherboard can handle, and you should be fine.

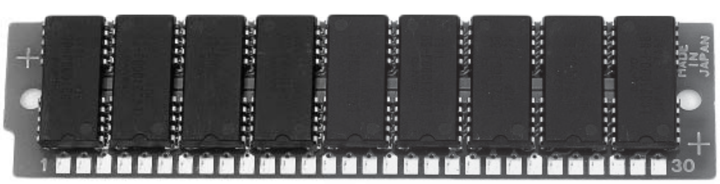
NOTE CAS stands for *column array strobe*, as mentioned earlier, one of the wires (along with the *row array strobe*) in the RAM that helps the memory controller find a particular bit of memory. Each of these wires requires electricity to charge up before it can do its job. This is one of the aspects of latency.

Parity and ECC

Given the high speeds and phenomenal amount of data moved by the typical DRAM chip, a RAM chip might occasionally give bad data to the memory controller. This doesn't necessarily mean that the RAM has gone bad. It could be a hiccup caused by some unknown event that makes a good DRAM chip say a bit is a zero when it's really a one. In most cases you won't even notice when such a rare event happens. In some environments, however, even these rare events are intolerable. A bank server handling thousands of online transactions per second, for example, can't risk even the smallest error. These important computers need a more robust, fault-resistant RAM.

The first type of error-detecting RAM was known as parity RAM (see Figure 5-19). *Parity RAM* stored an extra bit of data (called the parity bit) that the MCC used to verify whether the data was correct. Parity wasn't perfect. It wouldn't always detect an error, and if the MCC did find an error, it couldn't correct the error. For years, parity was the only available way to tell if the RAM made a mistake.

Figure 5-19
Ancient parity RAM stick



Today's PCs that need to watch for RAM errors use a special type of RAM called *error correction code RAM (ECC RAM)*. ECC is a major advance in error checking on DRAM. First, ECC detects any time a single bit is incorrect. Second, ECC fixes these errors on-the-fly. The checking and fixing come at a price, however, as ECC RAM is always slower than non-ECC RAM.

ECC DRAM comes in every DIMM package type and can lead to some odd-sounding numbers. You can find DDR2, DDR3, or DDR4 RAM sticks, for example, that come in 240-pin, *72*-bit versions. Similarly, you'll see 200-pin, 72-bit SO-DIMM format. The extra 8 bits beyond the 64-bit data stream are for the ECC.

You might be tempted to say, "Gee, maybe I want to try this ECC RAM." Well, don't! To take advantage of ECC RAM, you need a motherboard with an MCC designed to use ECC. Only expensive motherboards for high-end systems use ECC. The special-use-only nature of ECC makes it fairly rare. Plenty of techs with years of experience have never even seen ECC RAM.

NOTE Some memory manufacturers call the technology *error checking and correction* (*ECC*). Don't be thrown off if you see the phrase—it's the same thing, just a different marketing slant for error correction code.

Registered and Buffered Memory

When shopping for memory, especially for ECC memory, you are bound to come across the terms *registered RAM* or *buffered RAM* (CompTIA uses the latter term on the A+ 901 exam). Either term refers to a small register installed on some memory modules to act as a buffer between the DIMM and the memory controller. This little extra bit of circuitry helps compensate for electrical problems that crop up in systems with lots of memory modules, such as servers.

The key thing to remember is that a motherboard will use either buffered or *unbuffered RAM* (that's typical consumer RAM), not both. If you a module of the wrong type in a system you are upgrading, the worst that will happen is a blank screen and a lot of head scratching.

Working with RAM

Whenever someone comes up to me and asks what single hardware upgrade they can make to improve their system performance, I always tell them the same thing—add more RAM. Adding more RAM can improve overall system performance, processing speed, and stability—if you get it right. Botching the job can cause dramatic system instability, such as frequent, random crashes and reboots. Every tech needs to know how to install and upgrade system RAM of all types.

To get the desired results from a RAM upgrade, you must first determine if insufficient RAM is the cause of system problems. Second, you need to pick the proper RAM for the system. Finally, you must use good installation practices. Always store RAM sticks in anti-static packaging whenever they're not in use, and use strict ESD handling procedures.

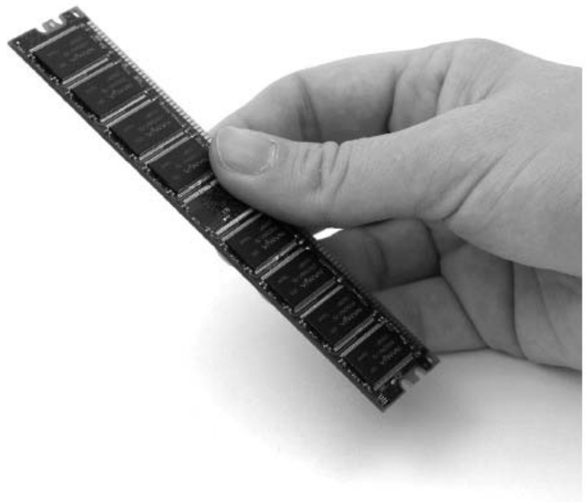


Figure 5-20
Don't do this! Grabbing the contacts is a bad idea!

Like many other pieces of the PC, RAM is *very* sensitive to ESD and other technician abuse (see Figure 5-20).

Do You Need More RAM?

Two symptoms point to the need for more RAM in a PC: general system sluggishness and excessive hard drive accessing. If programs take forever to load and running programs seem to stall and move more slowly than you would like, the problem could stem from insufficient RAM.

A friend with a Windows 7 system complained that her PC seemed snappy when she first got it but now takes a long time to do the things she wants to do with it, such as photograph retouching in Adobe Photoshop and document layout for an online magazine she produces. Her system had only 2 GB of RAM, sufficient to run Windows 7, but woefully insufficient for her tasks—she kept maxing out the RAM, and thus the system slowed to a crawl. I replaced her stick with a pair of 4-GB sticks and suddenly she had the powerhouse workstation she desired.

Excessive hard drive activity when you move between programs points to a need for more RAM. Every computer has the capability to make a portion of your hard drive work like RAM in case you run out of real RAM.

Virtual Memory

Computers use a portion of the hard drive as an extension of system RAM, through what's called virtual memory. Virtual memory is a portion of a hard drive or solid state drive set aside as what's called a *page file* or *swap file*. When a computer starts running out of real RAM because you've loaded too many programs, the system swaps programs from RAM to the page file, opening more space for programs currently active. All versions of

Windows, Mac OS X, and Linux use virtual memory. Let's use a typical Windows PC as an example of how paging works.

EXAM TIP The default and recommended page-file size in Windows is 1.5 times the amount of installed RAM on your computer.

Let's assume you have a PC with 4 GB of RAM. Figure 5-21 shows the system RAM as a thermometer with gradients from 0 to 4 GB. As programs load, they take up RAM, and as more and more programs are loaded (labeled A, B, and C in the figure), more RAM is used.

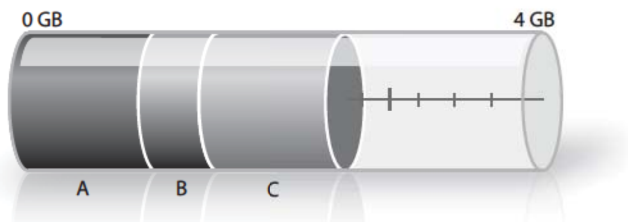


Figure 5-21 A RAM thermometer showing that more programs take more RAM

At a certain point, you won't have enough RAM to run any more programs (see Figure 5-22). Sure, you could close one or more programs to make room for yet another one, but you can't keep all of the programs running simultaneously. This is where virtual memory comes into play.

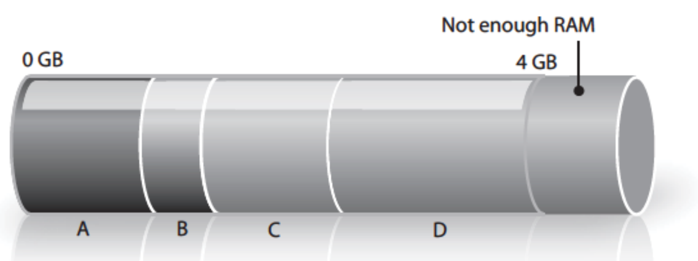


Figure 5-22 Not enough RAM to load program D

Windows' virtual memory starts by creating a page file that resides somewhere on your hard drive. The page file works like a temporary storage box. Windows removes running programs temporarily from RAM into the page file so other programs can load and run. If you have enough RAM to run all your programs, Windows does not need to use the page file—Windows brings the page file into play only when insufficient RAM is available to run all open programs.

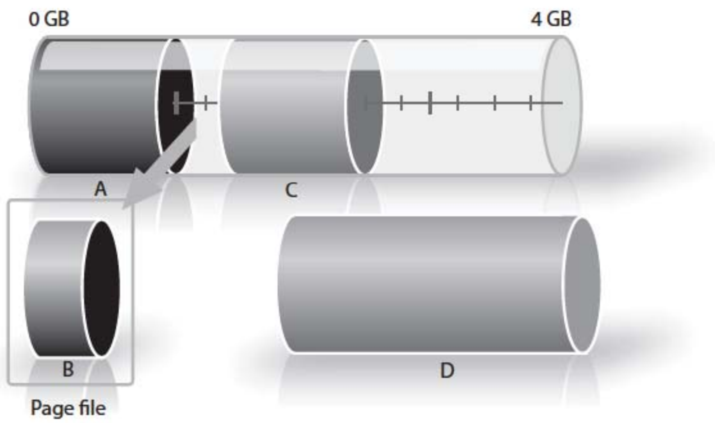


Figure 5-23 Program B being unloaded from memory

NOTE Virtual memory is a fully automated process and does not require any user intervention. This is true of virtual memory in Windows, Mac OS X, and Linux distributions ("distros").

To load, Program D needs a certain amount of free RAM. Clearly, this requires unloading some other program (or programs) from RAM without actually closing any programs. Windows looks at all running programs—in this case A, B, and C—and decides which program is the least used. That program is then cut out of or swapped from RAM and copied into the page file. In this case, Windows has chosen Program B (see Figure 5-23). Unloading Program B from RAM provides enough RAM to load Program D (see Figure 5-24).

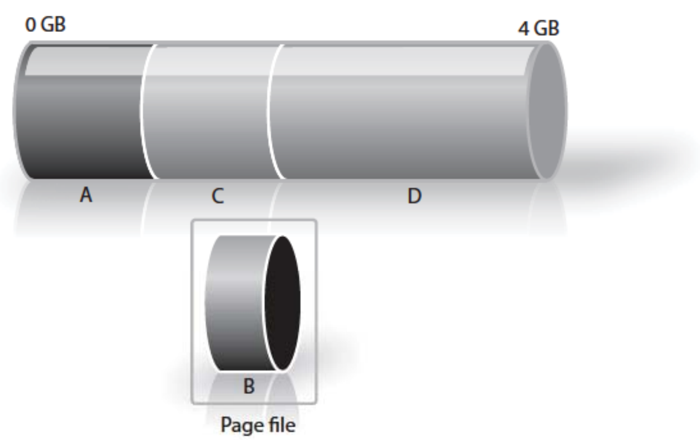


Figure 5-24 Program B stored in the page file, making room for Program D

It is important to understand that none of this activity is visible on the screen. Program B's window is still visible, along with those of all the other running programs. Nothing tells the user that Program B is no longer in RAM (see Figure 5-25).

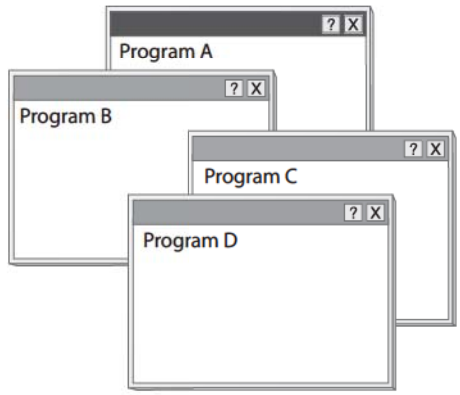


Figure 5-25 You can't tell whether a program is swapped or not.

So what happens if you click on Program B's window to bring it to the front? The program can't actually run from the page file; it must be loaded back into RAM. First, Windows decides which program must be removed from RAM, and this time Windows chooses Program C (see Figure 5-26). Then it loads Program B into RAM (see Figure 5-27).

Swapping programs to and from the page file and RAM takes time. Although no visual clues suggest that a swap is taking place, the machine slows down quite noticeably as Windows performs the swaps. Page files are a crucial aspect of Windows operation.

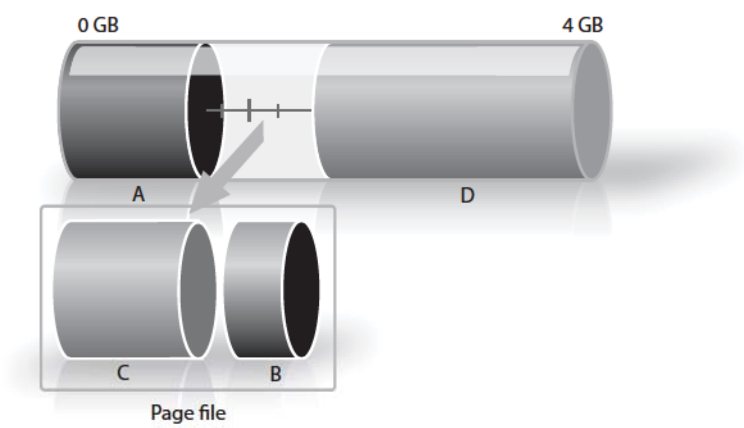


Figure 5-26 Program C is swapped to the page file.

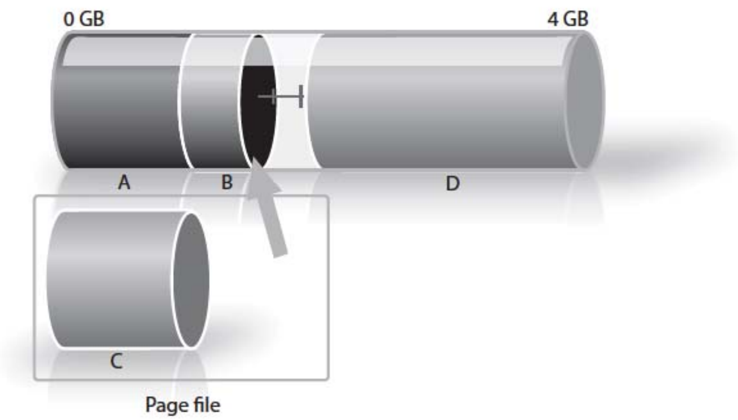


Figure 5-27 Program B is swapped back into RAM.

Windows handles page files automatically, but occasionally you'll run into problems and need to change the size of the page file or delete it and let Windows re-create it automatically. The page file is pagefile.sys. You can often find it in the root directory of the C: drive, but again, that can be changed. Wherever it is, the page file is a hidden system file, which means in practice that you'll have to play with your folder-viewing options to see it.

If Windows needs to access the page file too frequently, you will notice the hard drive access LED going crazy as Windows rushes to move programs between RAM and the page file in a process called *disk thrashing*. Windows uses the page file all the time, but excessive disk thrashing suggests that you need more RAM.

System RAM Recommendations

Microsoft sets very low minimum RAM requirements for the various Windows operating systems to get the maximum number of users to upgrade or convert, and that's fine. Old Windows machines ran well enough on 128 MB of RAM. Windows Vista raised the bar considerably, especially with the 64-bit versions of the operating system. Microsoft recommends a minimum system requirement of 1 GB of RAM for 32-bit versions of Windows and 2 GB of RAM for 64-bit versions. This applies to Windows Vista/7/8/8.1/10. I think that results in dreadfully sluggish computers. Here are my recommendations:

- **32-bit Windows** 2 GB to get by; 4 GB for best results
- **64-bit Windows** 4 GB to get by; 8 GB for a solid machine; 16+ GB for any machine doing serious, processor-intensive work

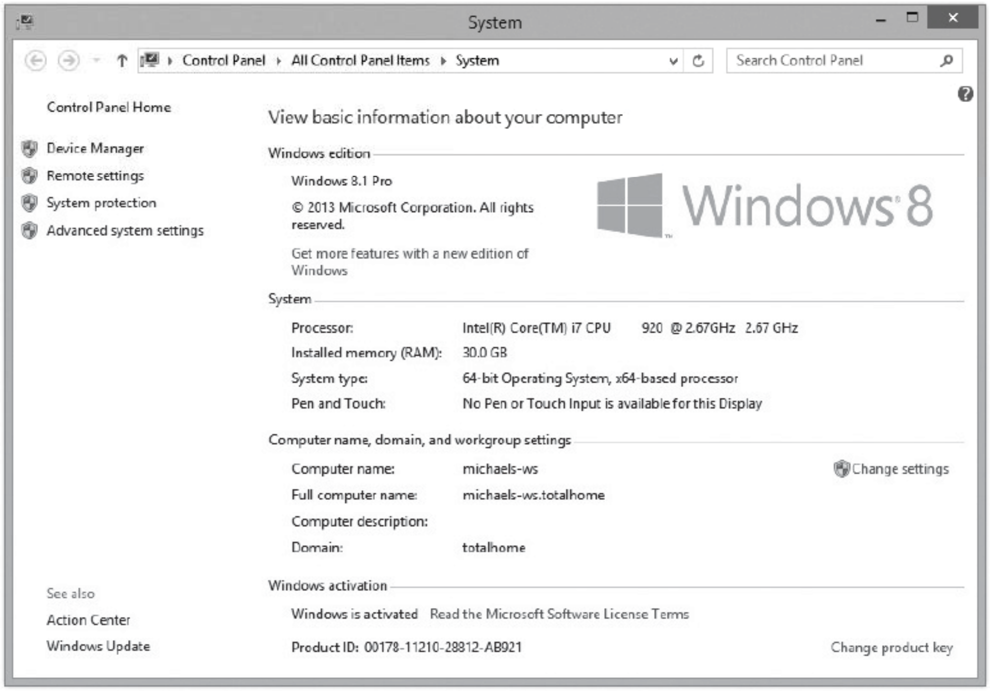


Figure 5-28 Mike has a lot of RAM!

The latest versions of Mac OS X require a minimum of 2 GB of RAM. Like Windows, however, the 64-bit-only OS does much better with a lot more RAM. I would go with 4 GB at a minimum, 8 GB for good performance, and more for peak performance.

Linux RAM requirements and recommendations depend entirely on the distro used. The mainstream ones, like Ubuntu, have requirements similar to Windows and Mac OS X. But many distros get by on very minimal system requirements.

Finally, a lot of personal computing devices have memory soldered in place and cannot be upgraded. That includes smartphones, tablets, and a host of portable computers, like the MacBook Air.

Determining Current RAM Capacity

Before you go get RAM, you obviously need to know how much RAM you currently have in your PC. Windows displays this amount in the System Control Panel applet (see Figure 5-28). You can also access the screen with the WINDOWS-PAUSE/BREAK keystroke combination.

Windows also includes the handy Performance tab in the Task Manager (as shown in Figure 5-29). The Performance tab includes a lot of information about the amount of RAM being used by your PC. Access the Task Manager by pressing CTRL-SHIFT-ESC and selecting the Performance tab.

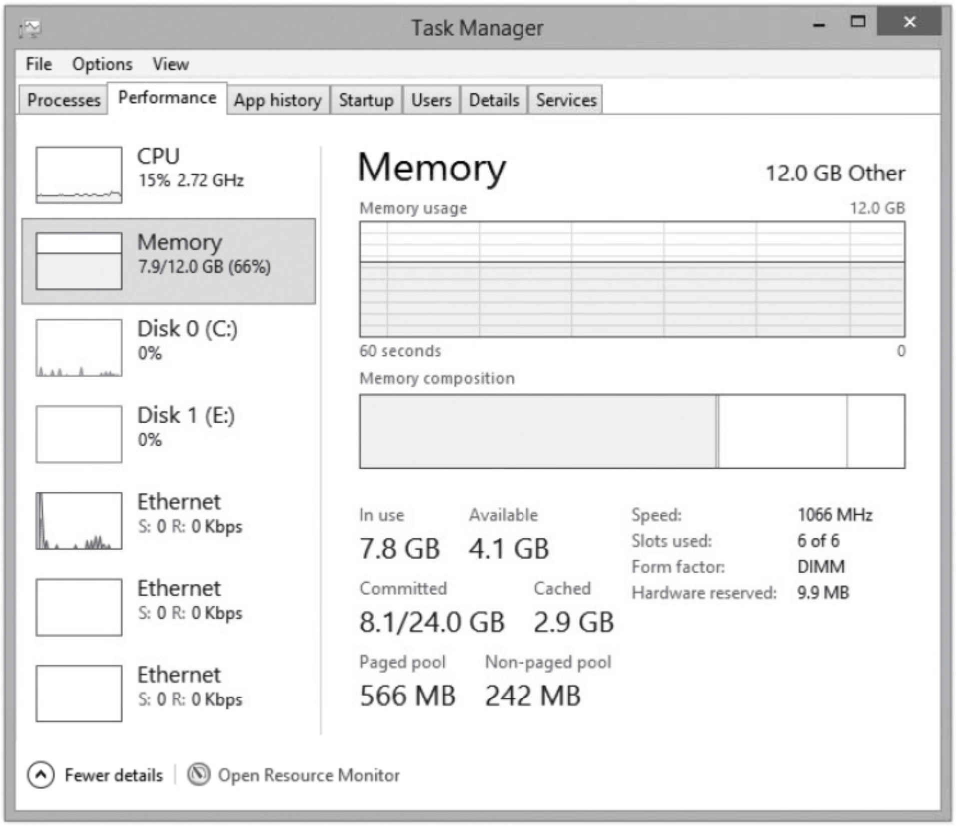


Figure 5-29 Performance tab in Windows 8.1 Task Manager

ReadyBoost

Windows Vista and later versions offer a feature called *ReadyBoost* that enables you to use flash media devices—removable USB thumb drives or memory cards—as super-fast, dedicated virtual memory. The performance gain over using only a typical hard drive for virtual memory can be significant with ReadyBoost because read/write access times on flash memory blows hard drive read/write access times away. Plus, the added ReadyBoost device or devices means Windows has multiple sources of virtual memory that it can use at the same time.

Windows 7 and later can handle up to eight flash devices, whereas Windows Vista can benefit from only one device. Devices can be between 1 and 32 GB in capacity. The flash device's file system matters in terms of how much memory Windows can use. Typically, the most you'll get out of a flash drive is 4 GB without manually changing the file system. Finally, Microsoft recommends using 1–3× the amount of system RAM for the ReadyBoost drives or devices to get optimal performance.

NOTE See Chapter 10, "Implementing Hard Drives," for an explanation of the differences between FAT, FAT32, NTFS, and exFAT.

Plug a ReadyBoost-approved device into a USB port or built-in flash memory card reader slot. Right-click the device in Computer and select Properties. Click the ReadyBoost tab and select the radio button next to either *Dedicate this device to ReadyBoost* or *Use this device* (see Figure 5-30). Click Apply to enhance your system's performance.

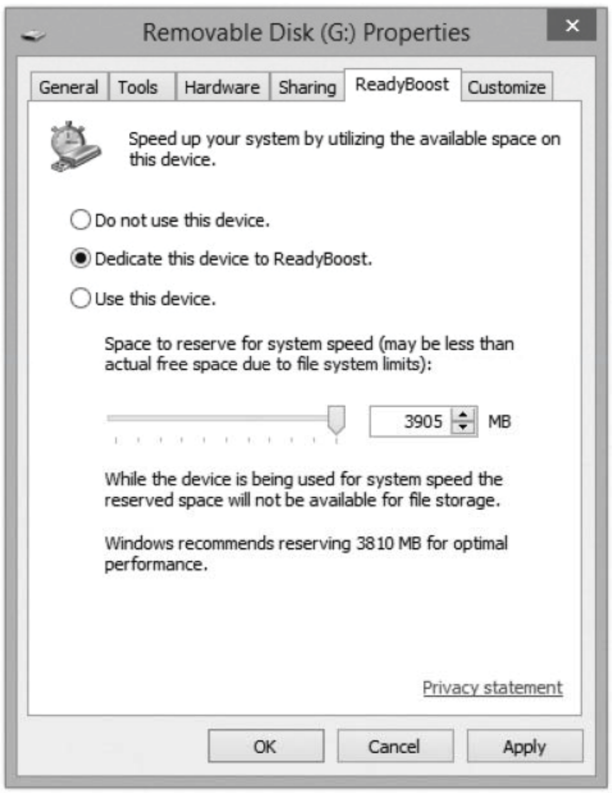


Figure 5-30 Dedicating a flash drive to ReadyBoost to enhance system performance

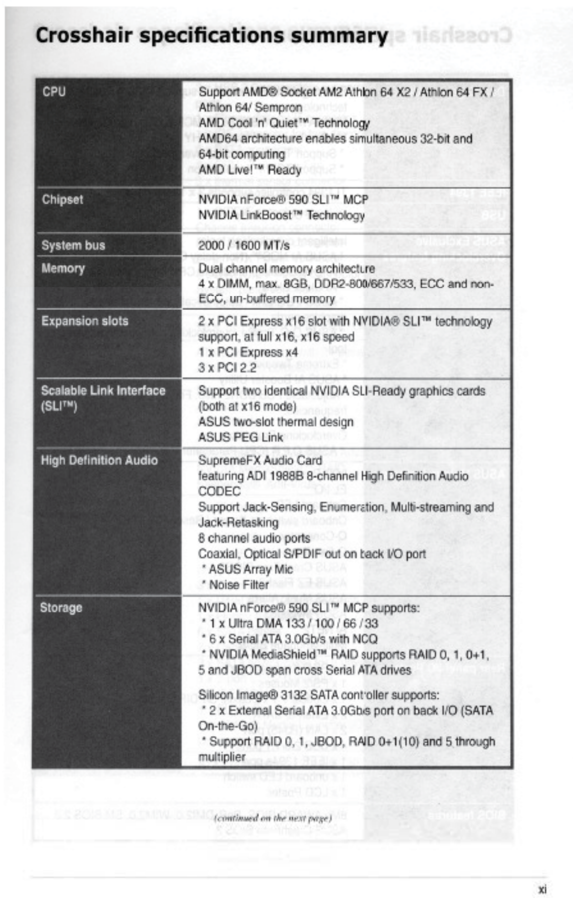
NOTE Adding more system memory will always give you or your clients a far better boost in performance than messing with ReadyBoost. ReadyBoost was essentially a bridge tool for Windows Vista adopters with marginal systems. RAM is relatively cheap today. Add some and blow off ReadyBoost. Just remember ReadyBoost for the exams.

Getting the Right RAM

To do the perfect RAM upgrade, determine the optimum amount of RAM to install and then get the right RAM for the motherboard. Your first two stops toward these goals are the inside of the case and your motherboard manual. Open the case to see how many sticks of RAM you have installed currently and how many free slots you have open.

Check the motherboard book or RAM manufacturer's Web site to determine the total capacity of RAM the system can handle and what specific technology works with your system.

You can't put DDR4 into a system that can only handle DDR3 SDRAM, after all, and it won't do you much good to install a pair of 4-GB DIMMs when your system tops out at 4 GB. Figure 5-31 shows the RAM limits for an ASUS Crosshair motherboard.



Crosshair specifications summary

CPU	Support AMD® Socket AM2 Athlon 64 X2 / Athlon 64 FX / Athlon 64/ Sempron AMD Cool 'n' Quiet™ Technology AMD64 architecture enables simultaneous 32-bit and 64-bit computing AMD Live!™ Ready
Chipset	NVIDIA nForce® 590 SLI™ MCP NVIDIA LinkBoost™ Technology
System bus	2000 / 1600 MT/s
Memory	Dual channel memory architecture 4 x DIMM, max. 8GB, DDR2-800/667/533, ECC and non-ECC, un-buffered memory
Expansion slots	2 x PCI Express x16 slot with NVIDIA® SLI™ technology support, at full x16, x16 speed 1 x PCI Express x4 3 x PCI 2.2
Scalable Link Interface (SLI™)	Support two identical NVIDIA SLI-Ready graphics cards (both at x16 mode) ASUS two-slot thermal design ASUS PEG Link
High Definition Audio	SupremeFX Audio Card featuring ADI 1988B 8-channel High Definition Audio CODEC Support Jack-Sensing, Enumeration, Multi-streaming and Jack-Retasking 8 channel audio ports Coaxial, Optical S/PDIF out on back I/O port * ASUS Array Mic * Noise Filter
Storage	NVIDIA nForce® 590 SLI™ MCP supports: * 1 x Ultra DMA 133 / 100 / 66 / 33 * 6 x Serial ATA 3.0Gb/s with NCQ * NVIDIA MediaShield™ RAID supports RAID 0, 1, 0+1, 5 and JBOD span cross Serial ATA drives Silicon Image® 3132 SATA controller supports: * 2 x External Serial ATA 3.0Gb/s port on back I/O (SATA On-the-Go) * Support RAID 0, 1, JBOD, RAID 0+1(10) and 5 through multiplier

(continued on the next page)

xi

Figure 5-31 The motherboard book shows how much RAM the motherboard will handle.

NOTE The freeware CPU-Z program tells you the total number of slots on your motherboard, the number of slots used, and the exact type of RAM in each slot—very handy. CPU-Z not only determines the latency of your RAM but also lists the latency at a variety of motherboard speeds. The media accompanying this book has a copy of CPU-Z, so check it out or download it from www.cpuid.com.

Mix and Match at Your Peril

All motherboards can handle different capacities of RAM. If you have three slots, you may put a 2-GB stick in one and a 4-GB stick in the other with a high chance of success. To ensure maximum stability in a system, however, shoot for as close as you can get to uniformity of RAM. Choose RAM sticks that match in technology, capacity, and speed.

Mixing Speeds

With so many different DRAM speeds available, you may often find yourself tempted to mix speeds of DRAM in the same system. Although you may get away with mixing speeds on a system, the safest, easiest rule to follow is to use the speed of DRAM specified in the motherboard book, and make sure that every piece of DRAM runs at that speed. In a worst-case scenario, mixing DRAM speeds can cause the system to lock up every few seconds or every few minutes. You might also get some data corruption. Mixing speeds sometimes works fine, but don't do your tax return on a machine with mixed DRAM speeds until the system has proven to be stable for a few days. The important thing to note here is that you won't break anything, other than possibly data, by experimenting.

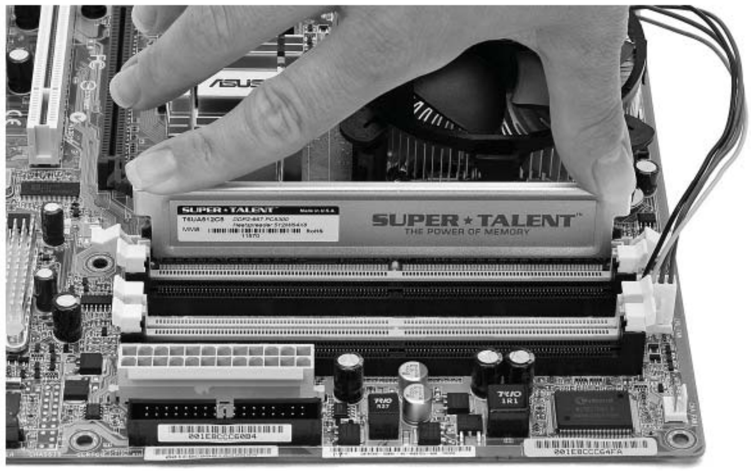
Okay, I have mentioned enough disclaimers. Modern motherboards provide some flexibility regarding RAM speeds and mixing. First, you can use RAM that is faster than the motherboard specifies. For example, if the system needs PC-3200 DDR2 SDRAM, you may put in PC-4200 DDR2 SDRAM and it should work fine. Faster DRAM is not going to make the system run any faster, however, so don't look for any system improvement.

Second, you can sometimes get away with putting one speed of DRAM in one bank and another speed in another bank, as long as all the speeds are as fast as or faster than the speed specified by the motherboard. Don't bother trying to put different-speed DRAM sticks in the same bank with a motherboard that uses dual-channel DDR.

Installing DIMMs

Installing DRAM is so easy that it's one of the very few jobs I recommend to non-techie folks. First, attach an anti-static wrist strap or touch some bare metal on the power supply to ground yourself and avoid ESD. Then swing the side tabs on the RAM slots down from the upright position. Pick up a stick of RAM—don't touch those contacts—and line up the notch or notches with the raised portion(s) of the DIMM socket (see Figure 5-32). A good hard push down is usually all you need to ensure a solid connection. Make sure that the DIMM snaps into position to show it is completely seated. Also, notice that the two side tabs move in to reflect a tight connection.

Figure 5-32
Inserting a DIMM

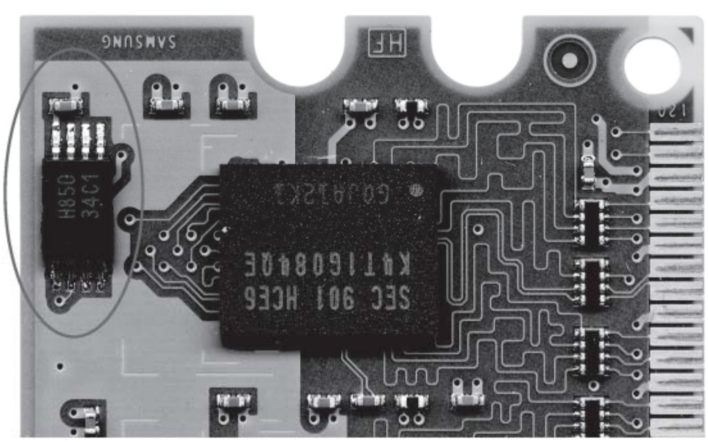


Serial Presence Detect (SPD)

Your motherboard should detect and automatically set up any DIMM you install, assuming you have the right RAM for the system, using a technology called *serial presence detect (SPD)*. RAM makers add a handy chip to modern sticks called the SPD chip (see Figure 5-33). The SPD chip stores all the information about your DRAM, including size, speed, ECC or non-ECC, registered or unregistered, and a number of other more technical bits of information.

When a PC boots, it queries the SPD chip so that the MCC knows how much RAM is on the stick, how fast it runs, and other information. Any program can query the SPD chip. Take a look at Figure 5-34 with the results of the popular CPU-Z program showing RAM information from the SPD chip.

Figure 5-33
SPD chip on a stick



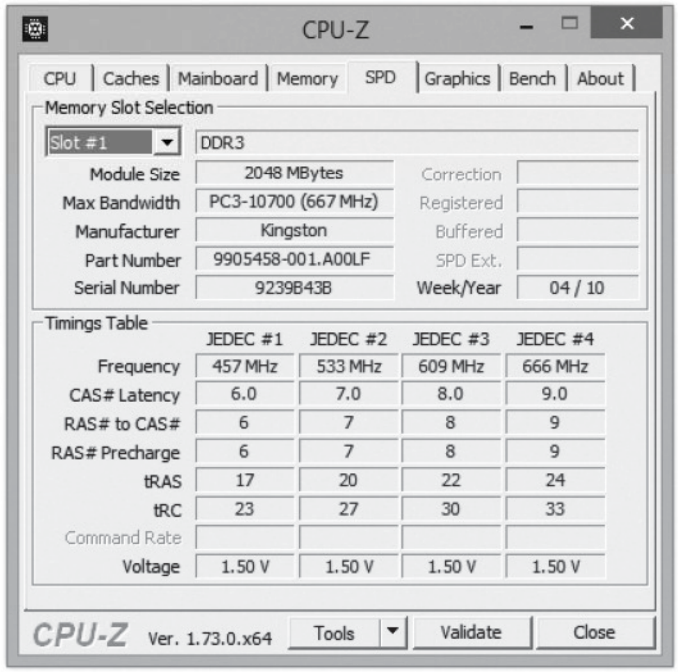


Figure 5-34 CPU-Z showing RAM information

All new systems count on SPD to set the RAM timings properly for your system when it boots. If you add a RAM stick with a bad SPD chip, you'll get a POST error message and the system will not boot. You can't fix a broken SPD chip; you just buy a new stick of RAM.

The RAM Count

Older systems display the RAM count during the initial boot sequence. After installing the new RAM, turn on the PC and watch the boot process closely. If you installed the RAM correctly, the RAM count on the PC reflects the new value (compare Figures 5.35 and 5.36). If the RAM value stays the same, you probably have installed the RAM in a slot the motherboard doesn't want you to use (for example, you may need to use a particular slot first) or have not installed the RAM properly. If the computer does not boot and you've got a blank screen, you probably have not installed all the RAM sticks correctly. Usually, a good second look is all you need to determine the problem. Reseat or reinstall the RAM stick and try again. RAM counts are confusing because RAM uses megabytes and gigabytes as opposed to millions and billions. Here are some examples of how different systems would show 256 MB of RAM:

268435456 (exactly 256 × 1 MB)

256M (some PCs try to make it easy for you)

262,144 (number of KB)

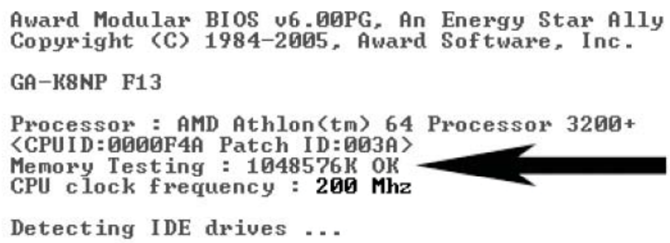


Figure 5-35
Hey, where's the rest of my RAM?!

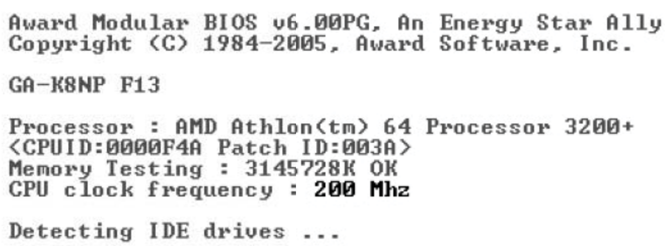


Figure 5-36
RAM count after proper insertion of DIMMs

You should know how much RAM you're trying to install and use some common sense. If you have 2 GB and you add another 2-GB stick, you should end up with 4 gigabytes of RAM. If you still see a RAM count of 2147483648 after you add the second stick, something went wrong!

Installing SO-DIMMs in Laptops

It wasn't that long ago that adding RAM to a laptop was either impossible or required you to send the system back to the manufacturer. For years, every laptop maker had custom-made, proprietary RAM packages that were difficult to handle and staggeringly expensive. The wide acceptance of SO-DIMMs has virtually erased these problems. Most laptops now provide relatively convenient access to their SO-DIMMs, enabling easy replacement or addition of RAM.

Access to RAM usually requires removing a panel or lifting up the keyboard—the procedure varies among laptop manufacturers. Figure 5-37 shows a typical laptop RAM access panel. You can slide the panel off to reveal the SO-DIMMs. Slide the pins into position and snap the SO-DIMM down into the retaining clips (see Figure 5-38).

Before doing any work on a laptop, turn the system off, disconnect it from the AC wall socket, and remove any removable batteries. Use an anti-static wrist strap because laptops are far more susceptible to ESD than desktop PCs.

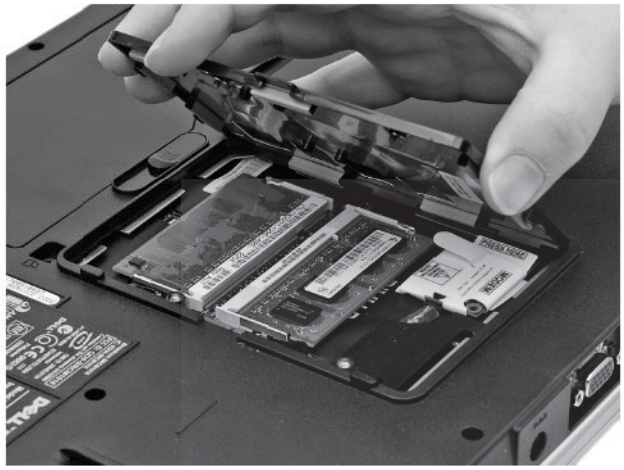


Figure 5-37 A RAM access panel on a laptop

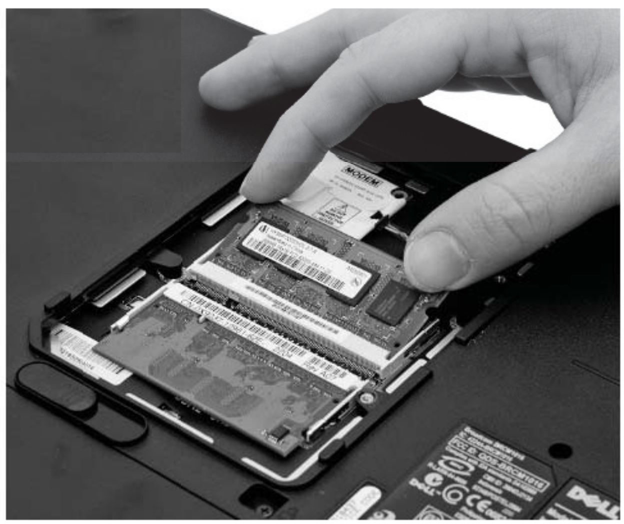


Figure 5-38 Snapping in a SO-DIMM

Troubleshooting RAM

"Memory" errors show up in a variety of ways on modern systems, including parity errors, ECC error messages, system lockups, page faults, and other error screens. These errors can indicate bad RAM but often point to something completely unrelated. This is especially true with intermittent problems. Techs need to recognize these errors and determine which part of the system caused the memory error.

You can get two radically different types of parity errors: real and phantom. *Real parity errors* are simply errors that the MCC detects from the parity or ECC chips (if you have them). The operating system then reports the problem in an error message, such as "Parity error at *xxxx:xxxxxxxx*," where *xxxx:xxxxxxxx* is a hexadecimal value (a string of numbers and letters, such as A5F2:004EEAB9). If you get an error like this, write down the value. A real parity/ECC error shows up at the same place in memory each time and almost always indicates that you have a bad RAM stick.

Phantom parity errors show up on systems that don't have parity or ECC memory. If Windows generates parity errors with different addresses, you most likely do *not* have a problem with RAM. These phantom errors can occur for a variety of reasons, including software problems, heat or dust, solar flares, fluctuations in the Force…you get the idea.

System lockups and page faults (they often go hand in hand) in Windows can indicate a problem with RAM. A system lockup is when the computer stops functioning. A *page fault* is a milder error that can be caused by memory issues but not necessarily system RAM problems. Certainly page faults *look* like RAM issues because Windows generates frightening error messages filled with long strings of hexadecimal digits, such as "KRNL386 caused a page fault at 03F2:25A003BC." Just because the error message contains a memory address, however, does not mean that you have a problem with your RAM. Write down the address. If it repeats in later error messages, you probably have a bad RAM stick. If Windows displays different memory locations, you need to look elsewhere for the culprit.

Every once in a while, something potentially catastrophic happens within the PC, some little electron hits the big red panic button, and the operating system has to shut down certain functions before it can save data. This panic button inside the PC is called a *non-maskable interrupt (NMI)*, more simply defined as an interruption the CPU cannot ignore. An NMI manifests as a *proprietary crash screen*. In Windows Vista and Windows 7, for example, the crash screen is what techs call the *Blue Screen of Death (BSoD)*—a bright blue screen with a scary-sounding error message on it (see Figure 5-39).

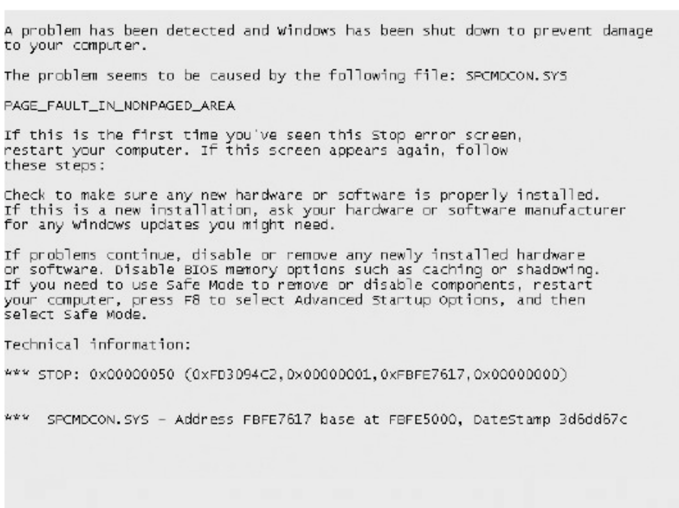


Figure 5-39 Blue Screen of Death

Windows 8/8.1/10 display a blue screen with a sad face and the words to the effect of Windows has a problem. Restart the machine. Mac OS X will display a spinning rainbow wheel called the *pinwheel of death*. The response is the same: reboot the machine.

Bad RAM sometimes triggers an NMI, although often the culprit lies with buggy programming or clashing code. The BSoD varies according to the operating system, and it would require a much lengthier tome than this one to cover all the variations. Suffice it to say that RAM *could* be the problem when that delightful blue screen appears.

Finally, intermittent memory errors can come from a variety of sources, including a dying power supply, electrical interference, buggy applications, buggy hardware, and so on. These errors show up as lockups, general protection faults, page faults, and parity errors, but they never have the same address or happen with the same applications. I always check the power supply first.

Testing RAM

Once you discover that you may have a RAM problem, you have a couple of options. First, several companies manufacture hardware RAM-testing devices. Second, you can use the method I use—*replace and pray*. Open the system case and replace each stick, one at a time, with a known good replacement stick. (You have one of those lying around, don't you?) This method, although potentially time-consuming, certainly works. With PC prices as low as they are now, you could simply replace the whole system for less than the price of a dedicated RAM tester.

Third, you could run a software-based tester on the RAM. Because you have to load a software tester into the memory it's about to scan, there's always a small chance that simply starting the software RAM tester might cause an error. Still, you can find some pretty good free ones out there. Windows 7 and later include the *Windows Memory Diagnostic* tool, which can automatically scan your computer's RAM when you encounter a problem. If you're using another OS, my favorite tool is memtest86+ (www.memtest.org). The memtest86+ software exhaustively checks your RAM and reports bad RAM when it finds it (see Figure 5-40).

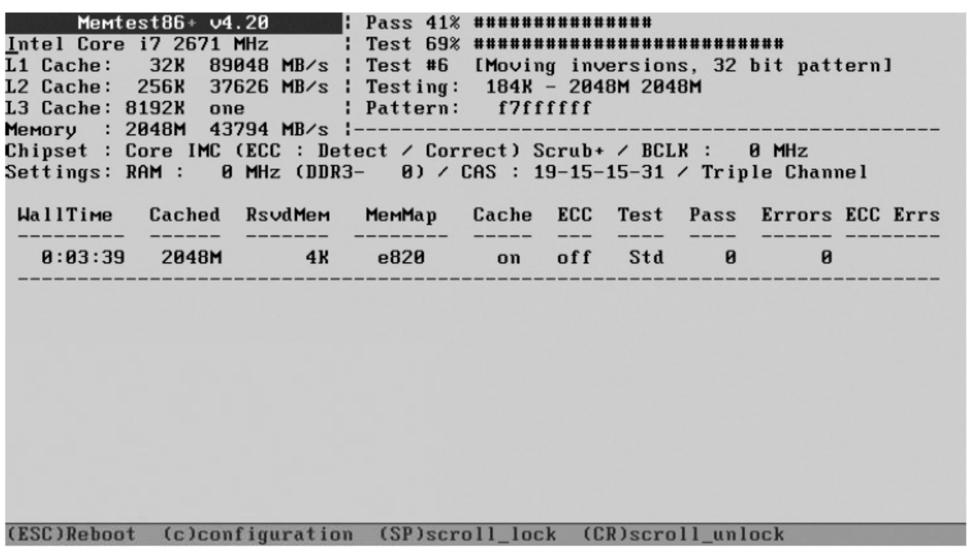


Figure 5-40 Memtest86+ in action

NOTE A *general protection fault (GPF)* is an error that can cause an application to crash. Often GPFs are caused by programs stepping on each other's toes. Chapter 17, "Troubleshooting Operating Systems," goes into more detail on GPFs and other Windows errors.

Chapter Review

Questions

1. Steve adds a second 1-GB 240-pin DIMM to his PC, which should bring the total RAM in the system up to 2 GB. The PC has an Intel Core 2 Duo 3-GHz processor and three 240-pin DIMM slots on the motherboard. When he turns on the PC, however, only 1 GB of RAM shows up during the RAM count. Which of the following is most likely to be the problem?
   A. Steve failed to seat the RAM properly.
   B. Steve put DDR SDRAM in a DDR 2 slot.
   C. The CPU cannot handle 2 GB of RAM.
   D. The motherboard can use only one RAM slot at a time.
2. Scott wants to add 512 MB of PC100 SDRAM to an aging but still useful desktop system. The system has a 100-MHz motherboard and currently has 256 MB of non-ECC SDRAM in the system. What else does he need to know before installing?
   A. What speed of RAM he needs.
   B. What type of RAM he needs.
   C. How many pins the RAM has.
   D. If the system can handle that much RAM.
3. What is the primary reason that DDR2 RAM is faster than DDR RAM?
   A. The core speed of the DDR2 RAM chips is faster.
   B. The input/output speed of the DDR2 RAM is faster.
   C. DDR RAM is single-channel and DDR2 RAM is dual-channel.
   D. DDR RAM uses 184-pin DIMMs and DDR2 uses 240-pin DIMMs.
4. What is the term for the delay in the RAM's response to a request from the MCC?
   A. Variance
   B. MCC gap
   C. Latency
   D. Fetch interval

5. How does an NMI manifest on a Mac OS X system?
   - **A.** Blue Screen of Death.
   - **B.** Pinwheel of death.
   - **C.** Interrupt of death.
   - **D.** NMIs only happen on Windows systems.
6. Silas has an AMD-based motherboard with two sticks of DDR2 RAM installed in two of the three RAM slots, for a total of 2 GB of system memory. When he runs CPU-Z to test the system, he notices that the software claims he's running single-channel memory. What could be the problem? (Select the best answer.)
   - **A.** His motherboard only supports single-channel memory.
   - **B.** His motherboard only supports dual-channel memory with DDR RAM, not DDR2.
   - **C.** He needs to install a third RAM stick to enable dual-channel memory.
   - **D.** He needs to move one of the installed sticks to a different slot to activate dual-channel memory.
7. Which of the following Control Panel applets will display the amount of RAM in your PC?
   - **A.** System
   - **B.** Devices and Printers
   - **C.** Device Manager
   - **D.** Action Center
8. What is the best way to determine the total capacity and specific type of RAM your system can handle?
   - **A.** Check the motherboard book.
   - **B.** Open the case and inspect the RAM.
   - **C.** Check the Device Manager.
   - **D.** Check the System utility in the Control Panel.
9. Gregor installed a third stick of known good RAM into his Core i7 system, bringing the total amount of RAM up to 3 GB. Within a few days, though, he started having random lockups and reboots, especially when doing memory-intensive tasks such as gaming. What is most likely the problem?
   - **A.** Gregor installed DDR RAM into a DDR2 system.
   - **B.** Gregor installed DDR2 RAM into a DDR3 system.
   - **C.** Gregor installed RAM that didn't match the speed or quality of the RAM in the system.
   - **D.** Gregor installed RAM that exceeded the speed of the RAM in the system.

10. Cindy installs a second stick of DDR3 RAM into her Core i5 system, bringing the total system memory up to 4 GB. Within a short period of time, though, she begins experiencing Blue Screens of Death. What could the problem be?
    - **A.** She installed faulty RAM.
    - **B.** The motherboard could only handle 2 GB of RAM.
    - **C.** The motherboard needed dual-channel RAM.
    - **D.** There is no problem. Windows always does this initially, but gets better after crashing a few times.

Answers

1. **A.** Steve failed to seat the RAM properly.
2. **D.** Scott needs to know if the system can handle that much RAM.
3. **B.** The input/output speed of DDR2 RAM is faster than that of DDR RAM (although the latency is higher).
4. **C.** Latency is the term for the delay in the RAM's response to a request from the MCC.
5. **B.** A non-maskable interrupt on a Mac OS X system often results in the spinning pinwheel called the pinwheel of death.
6. **D.** Motherboards can be tricky and require you to install RAM in the proper slots to enable dual-channel memory access. In this case, Silas should move one of the installed sticks to a different slot to activate dual-channel memory. (And he should check the motherboard manual for the proper slots.)
7. **A.** You can use the System applet to see how much RAM is currently in your PC.
8. **A.** The best way to determine the total capacity and specific type of RAM your system can handle is to check the motherboard book.
9. **C.** Most likely, Gregor installed RAM that didn't match the speed or quality of the RAM in the system.
10. **A.** If you have no problems with a system and then experience problems after installing something new, chances are the something new is at fault.