CHAPTER 8

Expansion Bus

In this chapter, you will learn how to

- Identify the structure and function of the expansion bus
- Describe the modern expansion bus
- Explain classic system resources
- Install expansion cards properly
- Troubleshoot expansion card problems

Expansion slots have been part of the PC from the very beginning. Way back then, IBM created the PC with an eye to the future; the original IBM PC had slots built into the motherboard—called *expansion slots*—for adding expansion cards and thus new functions to the PC. The slots and accompanying wires and support chips on the first PC and on the latest and greatest PC are called the *expansion bus*.

The expandability enabled by an expansion bus might seem obvious today, but think about the three big hurdles a would-be expansion card developer needed to cross to make a card that would work successfully in an expansion slot. First, any expansion card needed to be built specifically for the expansion slots—that would require the creation of industry standards. Second, the card needed some way to communicate with the CPU, both to receive instructions and to relay information. And third, the operating system would need some means of enabling the user to control the new device and thus take advantage of its functions. Here's the short form of those three hurdles:

- Physical connection
- Communication
- Drivers

This chapter covers the expansion bus in detail, starting almost at the very beginning of the PC—not because the history of the PC is inherently thrilling, but rather because the way the old PCs worked affects the latest systems. Installation today remains very similar to installation in 1987 in that you must have a physical connection, communication, and drivers for the operating system. Taking the time to learn the old ways first most definitely helps you understand and implement current technology, terminology, and practices

Historical/Conceptual

Structure and Function of the Expansion Bus

As you've learned, every device in the computer—whether soldered to the motherboard or snapped into a socket—connects to the external data bus and the address bus. The expansion slots are no exception. They connect to the rest of the PC through the chipset. Exactly *where* on the chipset varies depending on the system. On some systems, the expansion slots connect to the Southbridge (Figure 8-1). On other systems, the expansion slots connect to the Northbridge (Figure 8-2). Finally, many systems have more than one type of expansion bus, with slots of one type connecting to the Northbridge and slots of another type connecting to the Southbridge (Figure 8-3).

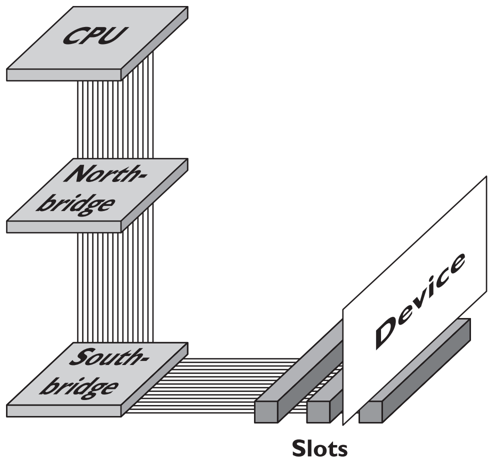


Figure 8-1 Expansion slots connecting to Southbridge

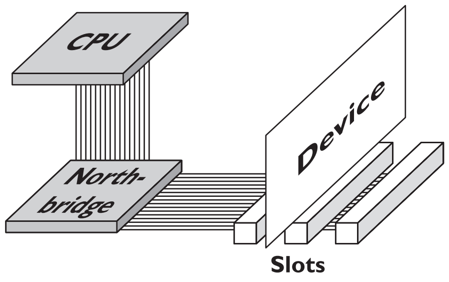


Figure 8-2 Expansion slots connecting to Northbridge

The chipset provides an extension of the address bus and data bus to the expansion slots, and thus to any expansion cards in those slots. If you plug a hard drive controller card into an expansion slot, it functions just as if it were built into the motherboard, albeit with one big difference: speed. As you'll recall from Chapter 5, "Microprocessors,"

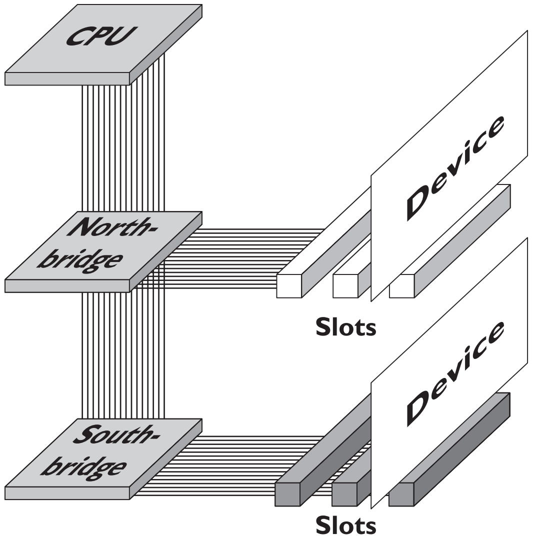


Figure 8-3
Expansion slots connecting to both Northbridge and Southbridge

the system crystal—the clock—pushes the CPU. The system crystal provides a critical function for the entire PC, acting like a drill sergeant calling a cadence, setting the pace of activity in the computer. Every device soldered to the motherboard is designed to run at the speed of the system crystal. A 133-MHz motherboard, for example, has at least a 133-MHz Northbridge chip and a 133-MHz Southbridge, all timed by a 133-MHz crystal (Figure 8-4).

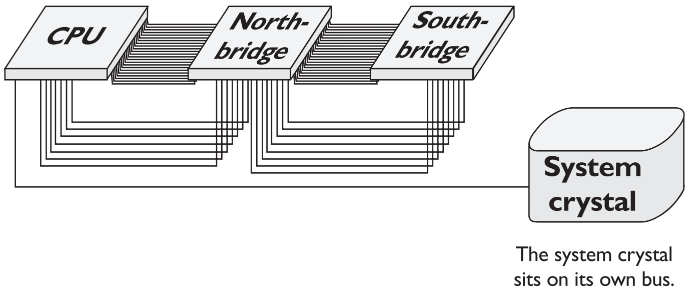


Figure 8-4
The system crystal sets the speed.

Clock crystals aren't just for CPUs and chipsets. Pretty much every chip in your computer has a CLK wire and needs to be pushed by a clock chip, including the chips on your expansion cards. Suppose you buy a device that did not come with your computer—say, a sound card. The chips on the sound card need to be pushed by a CLK signal from a crystal. If PCs were designed to use the system crystal to push that sound card, sound card manufacturers would need to make sound cards for every possible motherboard speed. You would have to buy a 100-MHz sound card for a 100-MHz system or a 133-MHz sound card for a 133-MHz system.

That would be ridiculous, and IBM knew it when they designed the PC. They had to make an extension to the external data bus that *ran at its own standardized speed*. You would use this part of the external data bus to snap new devices into the PC. IBM achieved this goal by adding a different crystal, called the *expansion bus crystal*, which controlled the part of the external data bus connected to the expansion slots (Figure 8-5).

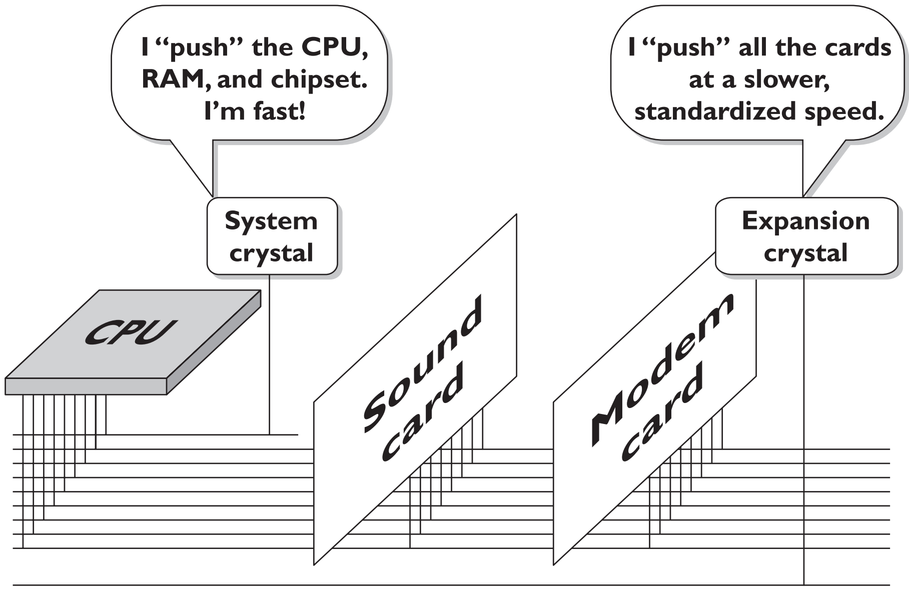

I "push" the CPU, RAM, and chipset. I'm fast!
I "push" all the cards at a slower, standardized speed.
System crystal
Expansion crystal
CPU
Sound card
Modem card

Figure 8-5 Function of system and expansion bus crystals

The expansion slots run at a much slower speed than the frontside bus. The chipset acts as the divider between the two buses, compensating for the speed difference with wait states and special buffering (storage) areas. No matter how fast the motherboard runs, the expansion slots run at a standard speed. In the original IBM PC, that speed was about 14.318 MHz ÷ 2, or about 7.16 MHz. The latest expansion buses run much faster, but remember that old speed of roughly 7 MHz; as you learn more about expansion slots, you'll see that it's still needed on even the most modern systems.

PC Bus

On first-generation IBM PCs, the 8088 CPU had an 8-bit external data bus and ran at a top speed of 4.77 MHz. IBM made the expansion slots on the first PCs with an 8-bit external bus connection. IBM wanted the bus to run as fast as the CPU, and even way back then, 4.77 MHz was an easy speed to achieve. IBM settled on a standard expansion bus speed of about 7 MHz—faster than the CPU! (This was the only occurrence in the history of PCs when the expansion bus was faster than the CPU.) This expansion bus was called the *PC bus* or *XT bus*. Figure 8-6 shows these ancient 8-bit expansion slots.

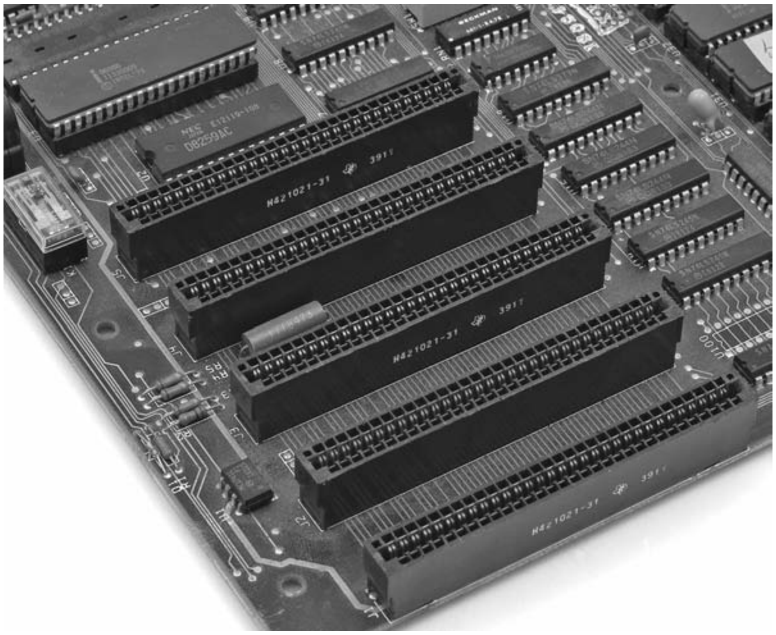


Figure 8-6 Eight-bit PC/XT slots

IBM certainly didn't invent the idea of the expansion bus—plenty of earlier computers, including many mainframes, had expansion slots—but IBM did something no one had ever done. They allowed competitors to copy the PC bus and make their own PCs without having to pay a licensing or royalty fee. They also allowed third parties to make cards that would snap into their PC bus. Remember that IBM invented the PC bus—it was (and still is) a patented product of IBM Corporation. By allowing everyone to copy the PC expansion bus technology, however, IBM established the industry standard and fostered the emergence of the clone market. If IBM had not allowed others to copy their patented technologies for free, companies such as Compaq, Dell, and Gateway never would have existed. Equally, component makers such as Logitech, Creative, and 3Com would never be the companies they are today without the help of IBM. Who knows? If IBM had not opened the PC bus to the world, this book and the A+ Certification exams might have been based on Apple computers.

PC Bus

8 bits wide

7-MHz speed

Manual configuration

ISA

When Intel invented the 286 processor, IBM wanted to create a new expansion bus that took advantage of the 286's 16-bit external data bus, yet also supported 8-bit cards. IBM achieved this by simply adding a set of connections to the end of the PC bus, creating a new 16-bit bus (Figure 8-7). Many techs called this bus the *AT bus* after the first system to use these slots, the 286-based IBM Advanced Technology (AT) computer. The AT bus ran at the same speed (approximately 7 MHz) as the earlier PC bus.

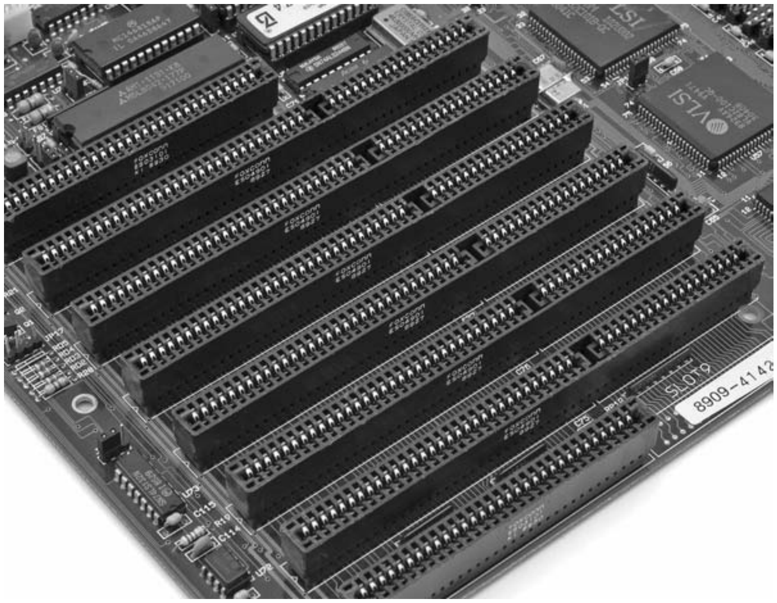


Figure 8-7 Sixteen-bit ISA or AT slots

Even though IBM allowed third parties to copy the PC and AT expansion bus architecture, they never released the complete specifications for these two types of expansion buses. In the early 1980s, a number of clone makers pooled their combined knowledge of the PC/XT and AT buses to create the *Industry Standard Architecture* (*ISA*).

The ISA bus enabled manufacturers to jump the first of the three hurdles for successful expansion cards, namely connectivity. If a company wanted to build a new kind of adapter card for the PC, they simply followed the specifications in the ISA standard.

ISA Bus

16 bits wide

7-MHz speed

Manual configuration

Essentials

Modern Expansion Buses

The ISA expansion bus was both excellent and cutting edge for its time, and was *the* expansion bus in every PC for the first ten years of the PC's existence. Yet ISA suffered from three tremendous limitations that began to cause serious bottlenecks by the late 1980s. First, ISA was slow, running at only about 7 MHz. Second, ISA was narrow—only 16 bits wide—and therefore unable to handle the 32-bit and 64-bit external data buses of more modern processors. Finally, techs had to configure ISA cards manually, making installation a time-consuming nightmare of running proprietary configuration programs and moving tiny jumpers just to get a single card to work.

Manufacturers clearly needed to come up with a better bus that addressed the many problems associated with ISA. They needed a bus that could take advantage of the 33-MHz motherboard speed and 32-bit-wide data bus found in 386 and 486 systems. They also wanted a bus that was self-configuring, freeing techs from the drudgery of manual configuration. Finally, they had to make the new bus backward compatible, so end users wouldn't have to throw out their oftentimes substantial investment in ISA expansion cards.

False Starts

In the late 1980s, several new expansion buses designed to address these shortcomings appeared on the market. Three in particular—IBM's Micro Channel Architecture (MCA), the open standard Extended ISA (EISA), and the Video Electronics Standards Association's VESA Local Bus (VL-Bus)—all had a few years of modest popularity from the late 1980s to the mid 1990s. Although all of these alternative buses worked well, they also had shortcomings that made them less than optimal replacements for ISA: IBM charged a heavy licensing fee for MCA, EISA was expensive to make, and VL-Bus only worked in tandem with the ISA bus. By 1993, the PC world was eager for a big name to come forward with a fast, wide, easy-to-configure, and cheap new expansion bus. Intel saw the need and stepped up to the plate with the now famous PCI bus.

PCI

Intel introduced the *peripheral component interconnect* (*PCI*) bus architecture (Figure 8-8) in the early 1990s, and the PC expansion bus was never again the same. Intel made many smart moves with PCI, not the least of which was releasing PCI to the public domain to make PCI very attractive to manufacturers. PCI provided a wider, faster, more flexible alternative than any previous expansion bus. The exceptional technology of the new bus, combined with the lack of a price tag, made manufacturers quickly drop ISA and the other alternatives and adopt PCI.

PCI really shook up the PC world with its capabilities. The original PCI bus was 32 bits wide and ran at 33 MHz, which was superb, but these features were expected and not earth-shattering. The coolness of PCI came from its capability to coexist with other

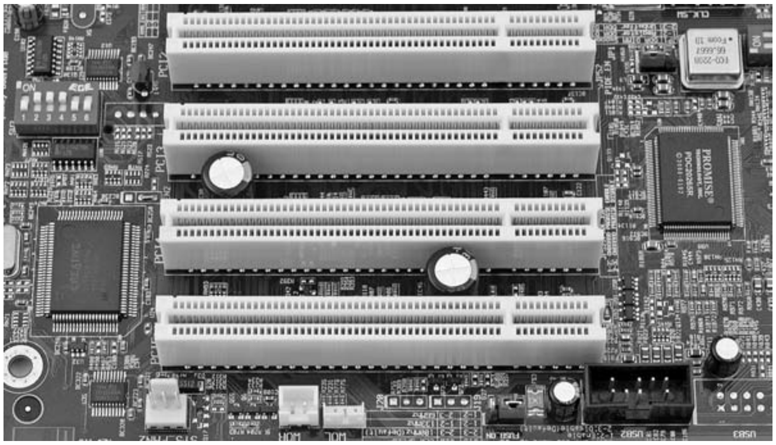


Figure 8-8 PCI expansion bus slots

expansion buses. When PCI first came out, you could buy a motherboard with both PCI and ISA slots. This was important because users could keep their old ISA cards and slowly migrate to PCI. Equally impressive was that PCI devices were (and still are) self-configuring, a feature that led to the industry standard that became known as plug and play. Finally, PCI had a powerful burst-mode feature that enabled very efficient data transfers.

NOTE Before PCI, it was rare to see more than one type of expansion slot on a motherboard. Today this is not only common—it's expected!

PCI Bus
32 bits wide
33-MHz speed
Self-configuring

TIP There was a 64-bit version of the original PCI standard, but it was quite rare.

The original PCI expansion bus has soldiered on in PCs for over ten years. Recently, more advanced forms have begun to appear. Although these new PCI expansion buses are faster than the original PCI, they're only improvements to PCI, not entirely new expansion buses. The original PCI might be fading away, but PCI in its many new forms is still "King of the Motherboard."

AGP

One of the big reasons for ISA's demise was video cards. When video started going graphical with the introduction of Windows, ISA buses were too slow and graphics looked terrible. PCI certainly improved graphics when it came out, but Intel was thinking ahead. Shortly after Intel invented PCI, they presented a specialized, video-only version of PCI called the *accelerated graphics port* (*AGP*). An AGP slot is a PCI slot, but one with a direct connection to the Northbridge. AGP slots are only for video cards—don't try to snap a sound card or modem into one. You'll learn much more about this fascinating technology in Chapter 19, "Video." Figure 8-9 shows a typical AGP slot.

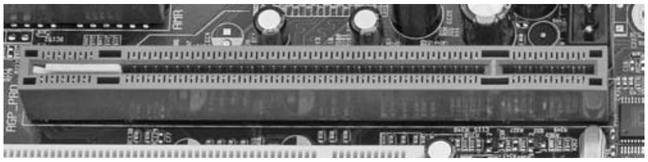


Figure 8-9
AGP slot

NOTE The AGP slot is almost universally brown in color, making it easy to spot.

PCI-X

PCI Extended (*PCI-X*), available in such systems as the Macintosh G5, is a huge enhancement to current PCI that is also fully backward compatible in terms of both hardware and software. PCI-X is a 64-bit-wide bus (see Figure 8-10). Its slots will accept regular PCI cards. The real bonus of PCI-X is its much enhanced speed. The PCI-X 2.0 standard features four speed grades (measured in MHz): PCI-X 66, PCI-X 133, PCI-X 266, and PCI-X 533.

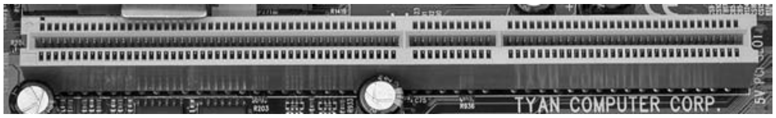


Figure 8-10
PCI-X slot

The obvious candidates for PCI-X are businesses using workstations and servers, because they have the "need for speed" and also the need for backward compatibility. Large vendors, especially in the high-end market, are already on board. HP, Dell, and Intel server products, for example, support PCI-X. A quick online shopping trip reveals tons of PCI-X stuff for sale: gigabit NICs, Fibre Channel cards, video adapters, and more.

Mini-PCI

PCI has even made it into laptops in the specialty *Mini-PCI* format (Figure 8-11). You'll find Mini-PCI in just about every laptop these days. Mini-PCI is designed to use low power and to lie flat—both good features for a laptop expansion slot. Mini-PCI returns in Chapter 21, "Portable Computing."

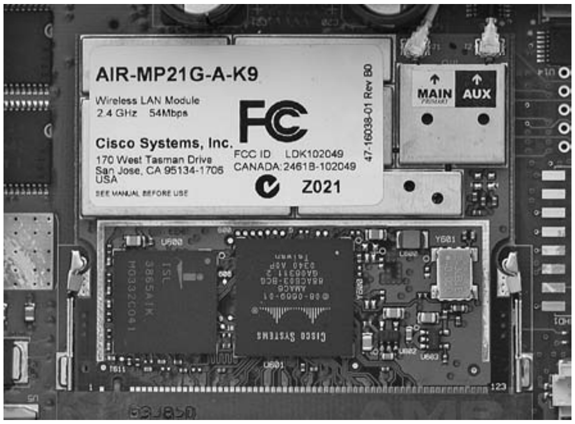


Figure 8-11 Tiny card in Mini-PCI slot. See the contacts at the bottom of the picture?

PCI Express

PCI Express (*PCIe*) is the latest, fastest, and most popular expansion bus in use today. As its name implies, PCI Express is still PCI, but it uses a point-to-point serial connection instead of PCI's shared parallel communication. Consider a single 32-bit chunk of data moving from a device to the CPU. In PCI parallel communication, 32 wires each carry one bit of that chunk of data. In serial communication, only one wire carries those 32 bits. You'd think that 32 are better than one, correct? Well, first of all, PCIe doesn't share the bus. A PCIe device has its own direct connection (a point-to-point connection) to the Northbridge, so it does not wait for other devices. Plus, when you start going really fast (think gigabits per second), getting all 32 bits of data to go from one device to another at the same time is difficult, because some bits get there slightly faster than others. That means you need some serious, high-speed checking of the data when it arrives to verify that it's all there and in good shape. Serial data doesn't have this problem, as all of the bits arrive one after the other in a single stream. When data is really going fast, a single point-to-point serial connection is faster than a shared 32-wire parallel connection.

And boy howdy, is PCIe ever fast! A PCIe connection uses one wire for sending and one for receiving. Each of these pairs of wires between a PCIe controller and a device is called a *lane*. Each direction of a lane runs at 2.5 Gbps, or 5 Gbps with PCIe 2.0. Better yet, each point-to-point connection can use 1, 2, 4, 8, 12, 16, or 32 lanes to achieve a maximum theoretical bandwidth of 320 Gbps. The effective data rate drops a little bit because of the *encoding scheme*—the way the data is broken down and reassembled—but full duplex data throughput can go up to a whopping 16 Gbps on a ×16 connection.
The most common PCIe slot is the 16-lane (×16) version most commonly used for video cards, as shown in Figure 8-12. The first versions of PCIe motherboards used a combination of a single PCIe ×16 slot and a number of standard PCI slots. (Remember, PCI is designed to work with other expansion slots, even other types of PCI.) There is also a small form factor version of PCI Express for mobile computers called PCI Express Mini Card.

NOTE When you talk about the lanes, such as ×1 or ×8, use "by" rather than "ex" for the multiplication mark. So "by 1" and "by 8" is the correct pronunciation. You'll of course hear it spoken as both "by 8" and "8 ex" for the next few years until the technology has become a household term.

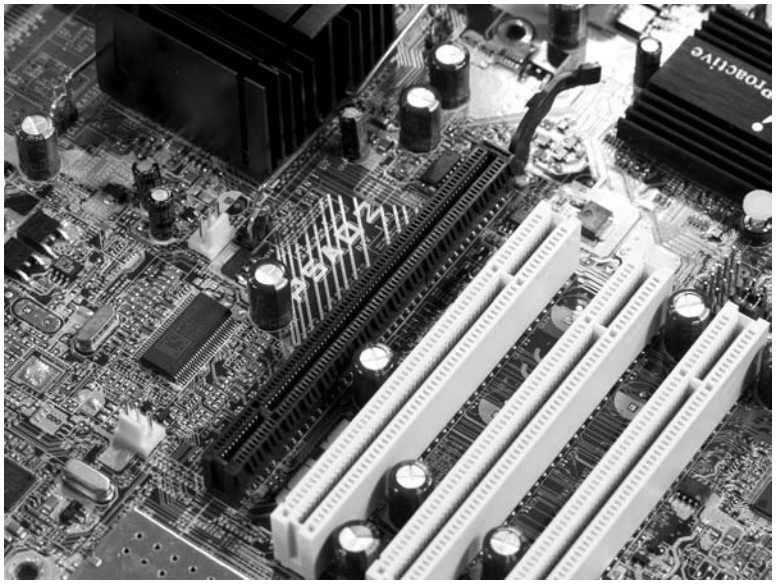


Figure 8-12
PCIe ×16 slot (black) with PCI slots (white)

The bandwidth generated by a ×16 slot is far more than anything other than a video card would need, so most PCIe motherboards also contain slots with fewer lanes. Currently ×1 and ×4 are the most common general-purpose PCIe slots, but PCIe is still pretty new—so expect things to change as PCIe matures (see Figure 8-13).

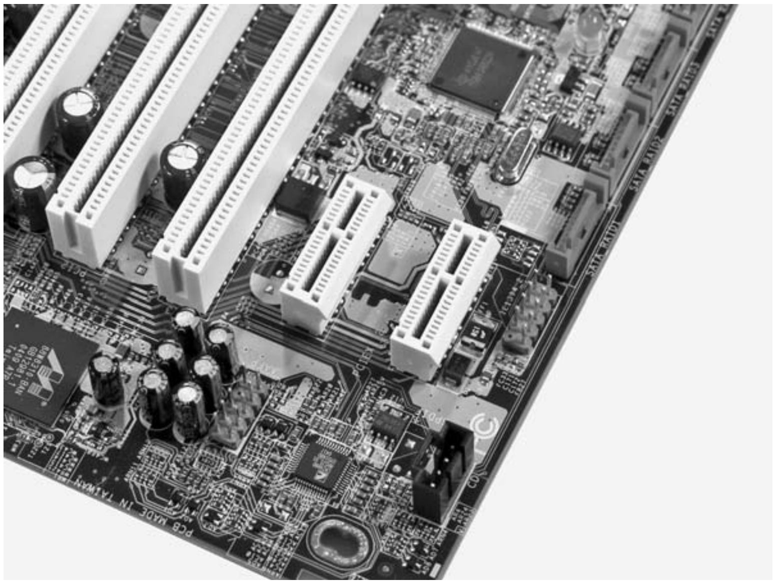


Figure 8-13
PCIe ×1 slots

System Resources

All devices on your computer, including your expansion cards, need to communicate with the CPU. Unfortunately, just using the word *communication* is too simplistic, because communication between the CPU and devices isn't like a human conversation. In the PC, only the CPU "talks" in the form of BIOS or driver commands—devices only react to the CPU's commands. You can divide communication into four aspects called *system resources*: I/O addresses, IRQs, DMA channels, and memory addresses.

Not all devices use all four system resources. All devices use I/O addressing and most use IRQs, but very few use DMA or memory. System resources are not new; they've been with PCs since the first IBM PC.

New devices must have their system resources configured. Configuration happens more or less automatically now through the plug and play process, but in the old days, configuration was handled through a painstaking manual process. (You kids don't know how good you have it. Oops! Sorry—Old Man Voice.) Even though system resources are now automated, you still might run into them in a few places on a modern PC. On those rare occasions, you'll need to understand I/O addresses, IRQs, DMAs, and memory to make changes as needed. Let's look at each system resource in detail to understand what they are and how they work.

I/O Addresses

The CPU gives a command to a device by using a pattern of ones and zeroes called an *I/O address*. Every device responds to at least four I/O addresses, meaning the CPU can give at least four different commands to each device. The process of communicating through I/O addresses is called, quite logically, *I/O addressing*. Here's how it works.

The chipset extends the address bus to the expansion slots, which makes two interesting things happen. First, you can place RAM on a card, and the CPU can address it just as it can your regular RAM. Devices such as video cards come with their own RAM. The CPU draws the screen by writing directly to the RAM on the video card. Second, the CPU can use the address bus to talk to all of the devices on your computer through I/O addressing.

Normally the address bus on an expansion bus works exactly like the address bus on a frontside bus—different patterns of ones and zeroes point to different memory locations. If the CPU wants to send an I/O address, however, it puts the expansion bus into what can be called I/O mode. When the bus goes into I/O mode, all devices on the bus look for patterns of ones and zeroes to appear on the address bus.

Back in the old Intel 8088 days, the CPU used an extra wire, called the *input/output or memory* (*IO/MEM*) *wire*, to notify devices that it was using the address bus either to specify an address in memory or to communicate with a particular device (Figure 8-14). You won't find an IO/MEM wire on a modern CPU, as the process has changed and become more complex—but the concept hasn't changed one bit. The CPU sends commands to devices by placing patterns of ones and zeroes—I/O addresses—on the address bus.

No two devices share the same I/O address because that would defeat the entire concept. To make sure no two devices share I/O addresses, all I/O addresses either are

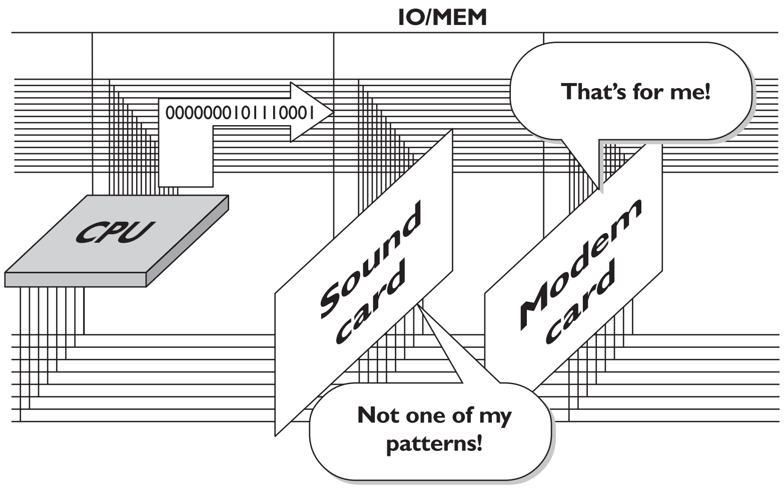
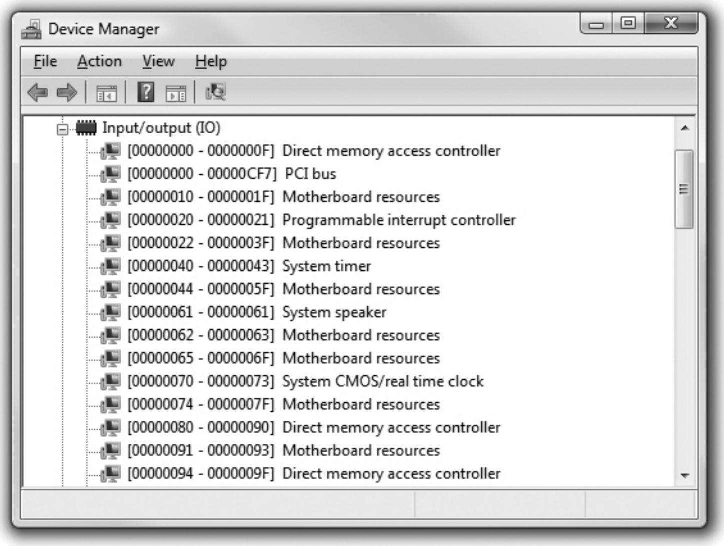


Figure 8-14 Sending out an I/O address

preset by standard (for example, all hard drive controllers use the same I/O addresses on every PC) or are set at boot by the operating system. You can see the I/O addresses for all of the devices on your computer by going into the Device Manager. Go to the View menu option and select *Resources by type*. Click on the plus sign directly to the left of the Input/output (IO) option to see a list of I/O addresses, as shown in Figure 8-15.



Input/output (IO)
[00000000 - 0000000F] Direct memory access controller
[00000000 - 00000CF7] PCI bus
[00000010 - 0000001F] Motherboard resources
[00000020 - 00000021] Programmable interrupt controller
[00000022 - 0000003F] Motherboard resources
[00000040 - 00000043] System timer
[00000044 - 0000005F] Motherboard resources
[00000061 - 00000061] System speaker
[00000062 - 00000063] Motherboard resources
[00000065 - 0000006F] Motherboard resources
[00000070 - 00000073] System CMOS/real time clock
[00000074 - 0000007F] Motherboard resources
[00000080 - 00000090] Direct memory access controller
[00000091 - 00000093] Motherboard resources
[00000094 - 0000009F] Direct memory access controller

Figure 8-15 Viewing resources by type, with I/O addresses expanded

Whoa! What's with all the letters and numbers? The address bus is always 32 bits (even if you have a 64-bit processor, the Northbridge only allows the first 32 bits to pass to the expansion slots), so instead of showing you the raw ones and zeroes, the Device Manager shows you the I/O address ranges in hexadecimal. Don't know hex? No worries—*hexadecimal* is just quick shorthand for representing the strings of ones and zeroes—*binary*—that you *do* know. One hex character is used to represent four binary characters. Here's the key:

0000 = 0

0001 = 1

0010 = 2

0011 = 3

0100 = 4

0101 = 5

0110 = 6

0111 = 7

1000 = 8

1001 = 9

1010 = A

1011 = B

1100 = C

1101 = D

1110 = E

1111 = F

Let's pick an arbitrary string of ones and zeroes:

    00000000000000000000000111110000

To convert to hex, just chop them into chunks of four:

    0000 0000 0000 0000 0000 0001 1111 0000

Then use the key above to convert:

    0 0 0 0 0 1 F 0

Then push the hex values together:

    000001F0

You now understand what those values mean in the Device Manager. Scroll down until you find the "[000001F0 – 000001F7] Primary IDE Channel" setting. Notice that two I/O addresses are listed. These show the entire range of I/O addresses for this device; the more complex the device, the more I/O addresses it uses. Address ranges are generally referred to by the first value in the range, commonly known as the *I/O base address*.

NOTE All I/O addresses only use the last 16 bits (they all start with 0000). Sixteen bits makes 2^{16} = 65,536 I/O address ranges—plenty for even the most modern PCs. Should PCs begin to need more I/O addresses in the future, the current I/O addressing system is ready.

Here are the most important items to remember about I/O addresses. First, every device on your PC has an I/O address. Without it, the CPU wouldn't have a way to send a device commands. Second, I/O addresses are configured automatically: you just plug in a device and it works. Third, no two devices should share I/O addresses. The system handles configuration, so this happens automatically.

Interrupt Requests

Between the standardized expansion bus connections and BIOS using I/O addressing, the CPU can now communicate with all of the devices inside the computer, but a third and final hurdle remains. I/O addressing enables the CPU to talk to devices, but how does a device tell the CPU it needs attention? How does the mouse tell the CPU that it has moved, for example, or how does the keyboard tell the CPU that somebody just pressed the J key? The PC needs some kind of mechanism to tell the CPU to stop doing whatever it is doing and talk to a particular device (Figure 8-16). This mechanism is called *interruption*.

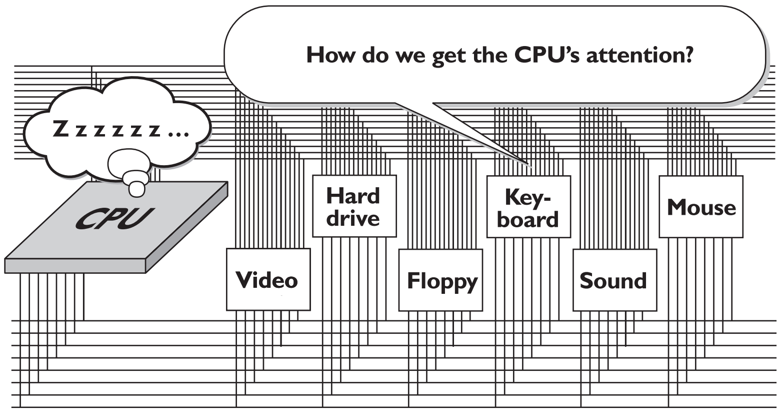


Figure 8-16 How do devices tell the CPU they need attention?

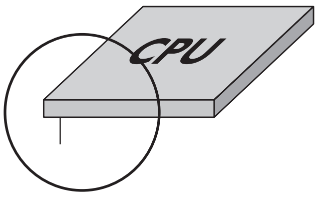


Figure 8-17
The INT wire

Every CPU in the PC world has an INT (interrupt) wire, shown in Figure 8-17. If a device puts voltage on this wire, the CPU stops what it's doing and deals with the interrupting device. Suppose you have a PC with only one peripheral, a keyboard that directly connects to the INT wire. If the user presses the J key, the keyboard charges the INT wire. The CPU temporarily stops running the browser (or whatever program is active) and runs the necessary BIOS routine to query the keyboard.

This would be fine if the computer had only one device. As you know, however, PCs have many devices, and almost all of them need to interrupt the CPU at some point. So the PC needs some kind of traffic cop to act as an intermediary between all of the devices and the CPU's INT wire. This traffic-cop chip, called the *I/O advanced programmable interrupt controller* (*IOAPIC*), uses special interrupt wires that run to all devices on the expansion bus (Figure 8-18).

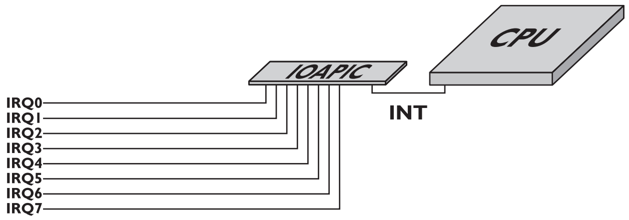


Figure 8-18
Eight interrupt wires (IRQs) run from the expansion bus to the IOAPIC.

NOTE IOAPIC functions are usually built into the Southbridge. Many developers drop the I/O part and simply call them *APICs*.

If a device wants to get the CPU's attention, it lights the interrupt wires with a special pattern of ones and zeroes just for that device. The IOAPIC then interrupts the CPU. The CPU queries the IOAPIC to see which device interrupted, and then it begins to communicate with the device over the address bus (Figure 8-19).

These unique patterns of ones and zeroes manifest themselves as something called *interrupt requests* (*IRQs*). Before IOAPICs, IRQs were actual wires leading to the previous generation of traffic cops, called PICs. It's easy to see if your system has a PIC or an IOAPIC. Go into the Device Manager and select Interrupt request (IRQ) with the resources set to show by type.

Figure 8-20 shows nearly a dozen IRQs, numbered 0 through 22, making this an IOAPIC system. IRQ 9 is special—this IRQ is assigned to the controller itself and is the IOAPIC's connection to the CPU. If you look closely, you'll also notice that some IRQs aren't listed. These are unused or "open" IRQs. If you add another device to the system,

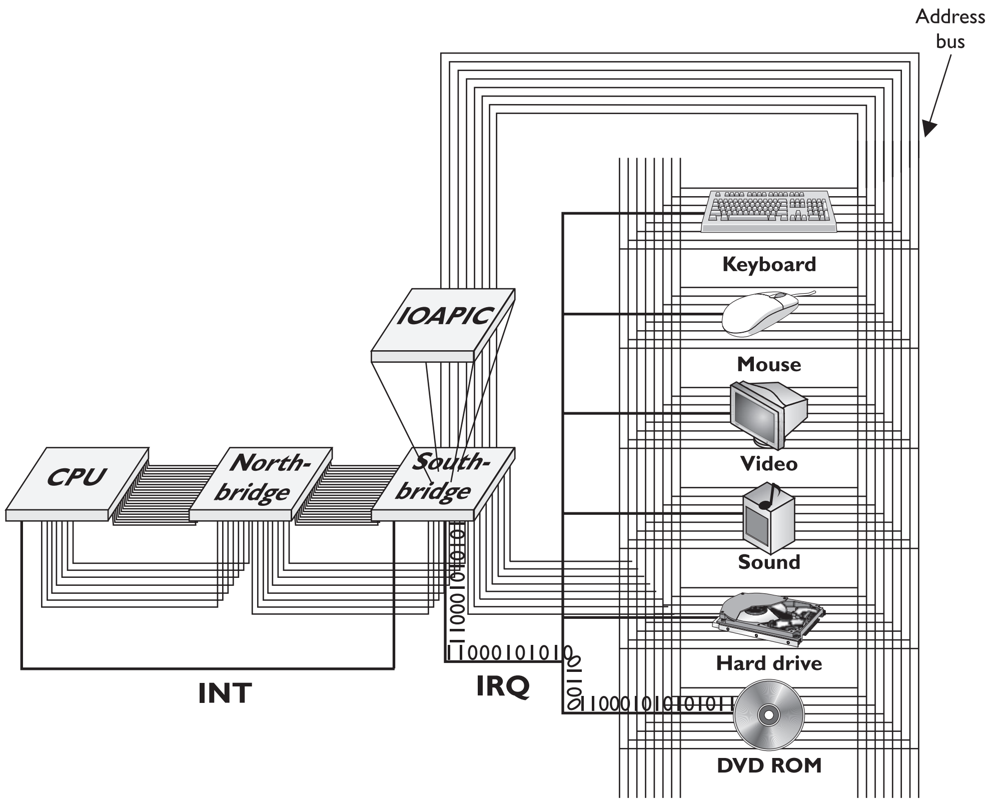


Figure 8-19 IOAPIC at work

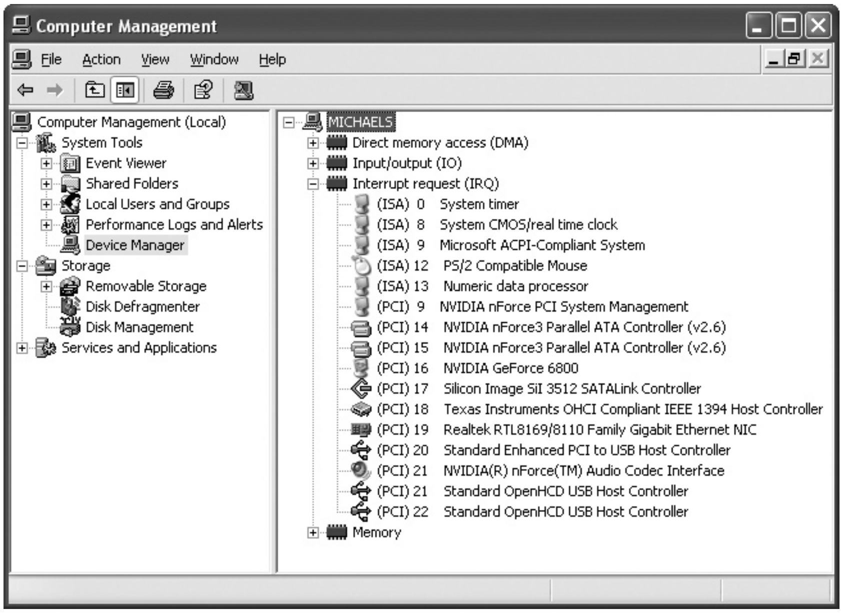
the new device will take up one of these unused IRQs. Now look at the older PIC system in Figure 8-21—note that it only shows IRQs ranging from 0 through 15.

Modern systems running Windows Vista or Windows 7 can use virtual IRQs to support devices. A quick glance at the Device Manager IRQ list in Vista shows a giant list of IRQs, ranging from 0 to upwards of 190 (Figure 8-22). Some systems even display negative IRQs, such as –2 or –4. Once you get into a Windows Vista or Windows 7 machine, you can pretty much forget about IRQs. The OS handles it all automatically.

Let's look at the last serious vestige of the "bad old days" on your PC: COM and LPT ports.

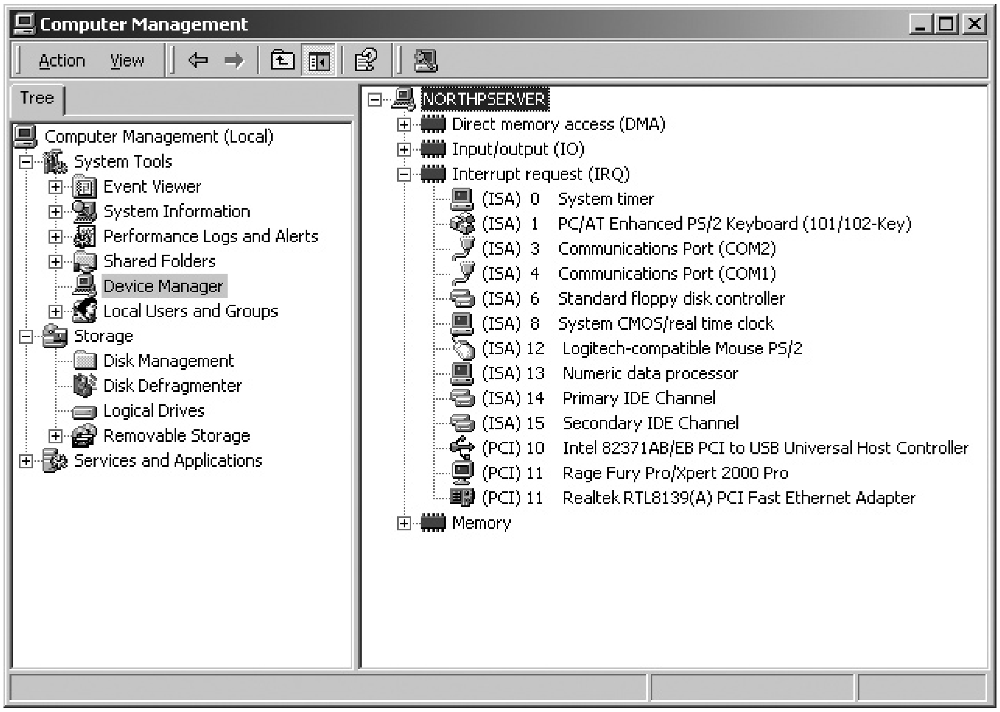
COM and LPT Ports

When the PC first came out, the I/O addresses and IRQ for every device had to be manually configured. How you did this varied from device to device: you moved jumpers, turned dials, or ran weird configuration programs. It was never easy. IBM tried to make configuration easier by creating preset I/O address and IRQ combinations for



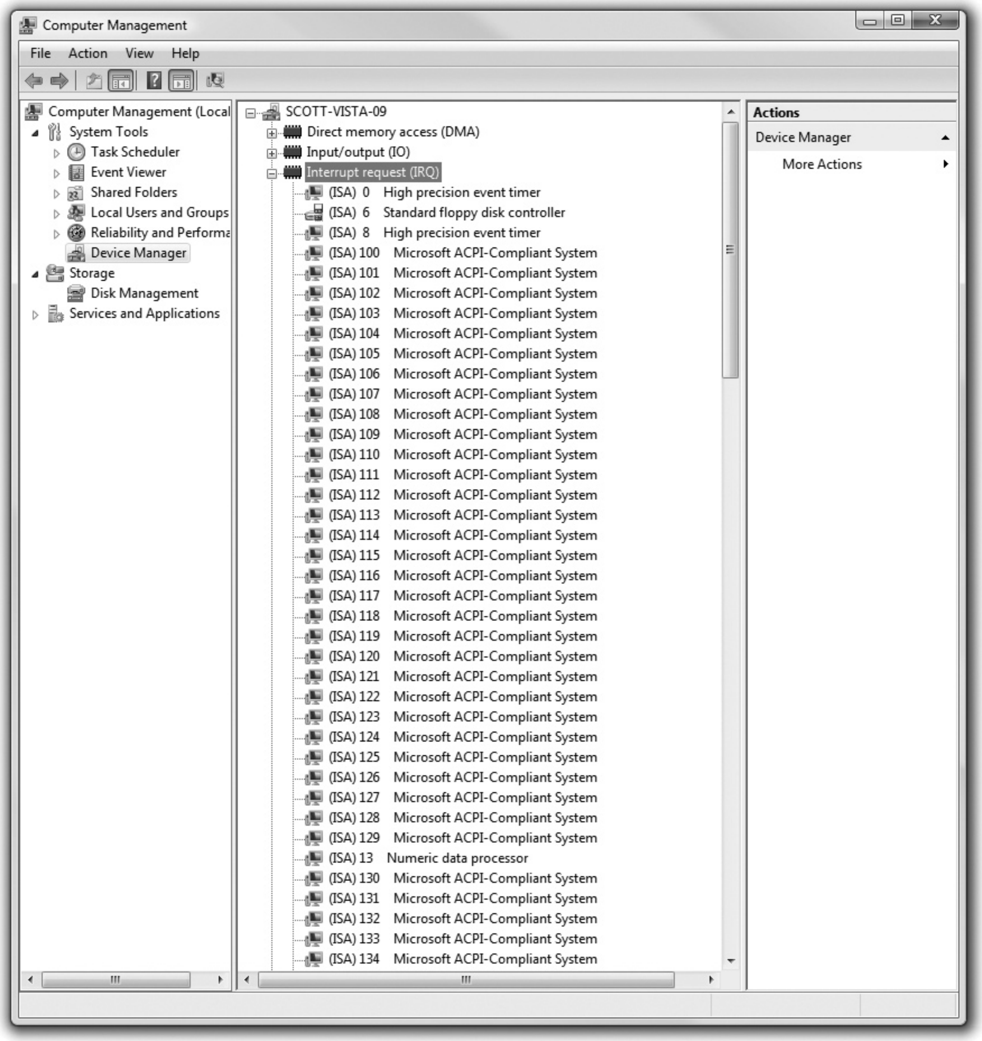
```
MICHAELS
  Direct memory access (DMA)
  Input/output (IO)
  Interrupt request (IRQ)
    (ISA) 0   System timer
    (ISA) 8   System CMOS/real time clock
    (ISA) 9   Microsoft ACPI-Compliant System
    (ISA) 12  PS/2 Compatible Mouse
    (ISA) 13  Numeric data processor
    (PCI) 9   NVIDIA nForce PCI System Management
    (PCI) 14  NVIDIA nForce3 Parallel ATA Controller (v2.6)
    (PCI) 15  NVIDIA nForce3 Parallel ATA Controller (v2.6)
    (PCI) 16  NVIDIA GeForce 6800
    (PCI) 17  Silicon Image SiI 3512 SATALink Controller
    (PCI) 18  Texas Instruments OHCI Compliant IEEE 1394 Host Controller
    (PCI) 19  Realtek RTL8169/8110 Family Gigabit Ethernet NIC
    (PCI) 20  Standard Enhanced PCI to USB Host Controller
    (PCI) 21  NVIDIA(R) nForce(TM) Audio Codec Interface
    (PCI) 21  Standard OpenHCD USB Host Controller
    (PCI) 22  Standard OpenHCD USB Host Controller
  Memory
```

Figure 8-20 IRQs in an IOAPIC system



```
NORTHPSERVER
  Direct memory access (DMA)
  Input/output (IO)
  Interrupt request (IRQ)
    (ISA) 0   System timer
    (ISA) 1   PC/AT Enhanced PS/2 Keyboard (101/102-Key)
    (ISA) 3   Communications Port (COM2)
    (ISA) 4   Communications Port (COM1)
    (ISA) 6   Standard floppy disk controller
    (ISA) 8   System CMOS/real time clock
    (ISA) 12  Logitech-compatible Mouse PS/2
    (ISA) 13  Numeric data processor
    (ISA) 14  Primary IDE Channel
    (ISA) 15  Secondary IDE Channel
    (PCI) 10  Intel 82371AB/EB PCI to USB Universal Host Controller
    (PCI) 11  Rage Fury Pro/Xpert 2000 Pro
    (PCI) 11  Realtek RTL8139(A) PCI Fast Ethernet Adapter
  Memory
```

Figure 8-21 IRQs in a PIC system



```
Computer Management (Local
  System Tools
    Task Scheduler
    Event Viewer
    Shared Folders
    Local Users and Groups
    Reliability and Performa
    Device Manager
  Storage
    Disk Management
  Services and Applications
```

```
SCOTT-VISTA-09
  Direct memory access (DMA)
  Input/output (IO)
  Interrupt request (IRQ)
    (ISA) 0    High precision event timer
    (ISA) 6    Standard floppy disk controller
    (ISA) 8    High precision event timer
    (ISA) 100  Microsoft ACPI-Compliant System
    (ISA) 101  Microsoft ACPI-Compliant System
    (ISA) 102  Microsoft ACPI-Compliant System
    (ISA) 103  Microsoft ACPI-Compliant System
    (ISA) 104  Microsoft ACPI-Compliant System
    (ISA) 105  Microsoft ACPI-Compliant System
    (ISA) 106  Microsoft ACPI-Compliant System
    (ISA) 107  Microsoft ACPI-Compliant System
    (ISA) 108  Microsoft ACPI-Compliant System
    (ISA) 109  Microsoft ACPI-Compliant System
    (ISA) 110  Microsoft ACPI-Compliant System
    (ISA) 111  Microsoft ACPI-Compliant System
    (ISA) 112  Microsoft ACPI-Compliant System
    (ISA) 113  Microsoft ACPI-Compliant System
    (ISA) 114  Microsoft ACPI-Compliant System
    (ISA) 115  Microsoft ACPI-Compliant System
    (ISA) 116  Microsoft ACPI-Compliant System
    (ISA) 117  Microsoft ACPI-Compliant System
    (ISA) 118  Microsoft ACPI-Compliant System
    (ISA) 119  Microsoft ACPI-Compliant System
    (ISA) 120  Microsoft ACPI-Compliant System
    (ISA) 121  Microsoft ACPI-Compliant System
    (ISA) 122  Microsoft ACPI-Compliant System
    (ISA) 123  Microsoft ACPI-Compliant System
    (ISA) 124  Microsoft ACPI-Compliant System
    (ISA) 125  Microsoft ACPI-Compliant System
    (ISA) 126  Microsoft ACPI-Compliant System
    (ISA) 127  Microsoft ACPI-Compliant System
    (ISA) 128  Microsoft ACPI-Compliant System
    (ISA) 129  Microsoft ACPI-Compliant System
    (ISA) 13   Numeric data processor
    (ISA) 130  Microsoft ACPI-Compliant System
    (ISA) 131  Microsoft ACPI-Compliant System
    (ISA) 132  Microsoft ACPI-Compliant System
    (ISA) 133  Microsoft ACPI-Compliant System
    (ISA) 134  Microsoft ACPI-Compliant System
```

Figure 8-22 IRQs in Windows Vista

the serial and parallel ports, because they were the most commonly used ports on the original PC. These preset combinations were called *COM ports* for serial connections and *LPT ports* for parallel ports. Table 8-1 lists the early preset combinations of I/O addresses and IRQs.

TIP The term "COM" for serial ports came from "communication," and the term "LPT" for parallel ports came from "line printer."

Table 8-1 COM and LPT Assignments

Port	I/O Base Address	IRQ
COM1	03F8	4
COM2	02F8	3
COM3	03E8	4
COM4	02E8	3
LPT1	0378	7
LPT2	0278	5

Notice that the four COM ports share two IRQs. In the old days, if two devices shared an IRQ, the system instantly locked up. The lack of available IRQs in early systems led IBM to double up the IRQs for the serial devices, creating one of the few exceptions to the rule that no two devices could share IRQs. You could share an IRQ between two devices, but only if one of the devices would never actually access the IRQ. You'd see this with a dedicated fax/modem card, for example, which has a single phone line connected to a single card that has two different functions. The CPU needed distinct sets of I/O addresses for fax commands and modem commands, but as there was only the one modem doing both jobs, it needed only a single IRQ.

Direct Memory Access

CPUs do a lot of work. They run the BIOS, operating system, and applications. CPUs handle interrupts and I/O addresses. CPUs also deal with one other item: data. CPUs constantly move data between devices and RAM. CPUs move files from the hard drive to RAM. They move print jobs from RAM to laser printers, and they move images from scanners to RAM, just to name a very few examples of this RAM-to-device-and-back process.

Moving all this data is obviously necessary, but it is a simple task—the CPU has better things to do with its power and time. Moreover, with all of the caches and such on today's CPUs, the system spends most of its time waiting around doing nothing while the CPU handles some internal calculation. Add these facts together and the question arises: Why not make devices that access memory directly, without involving the CPU (Figure 8-23)? The process of accessing memory without using the CPU is called *direct memory access* (*DMA*).

DMA is very common and is excellent for creating background sounds in games and for moving data from floppy and hard drives into RAM (Figure 8-24).

Nice as it may sound, the concept of DMA as just described has a problem—there's only one expansion bus. What if more than one device wants to use DMA? What keeps these devices from stomping on the external data bus all at the same time? Plus, what if the CPU suddenly needs the data bus? How can you stop the device using DMA so the CPU, which should have priority, can access the bus? To deal with this, IBM added another traffic cop.

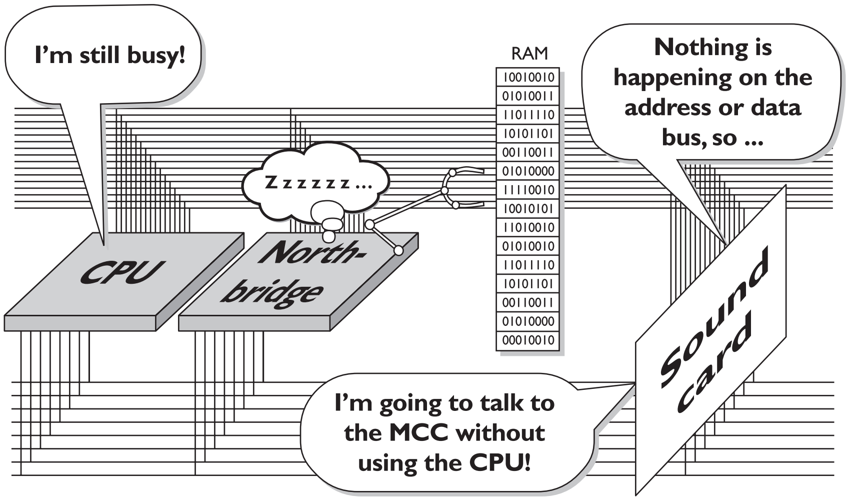


Figure 8-23 Why not talk to the chipset directly?

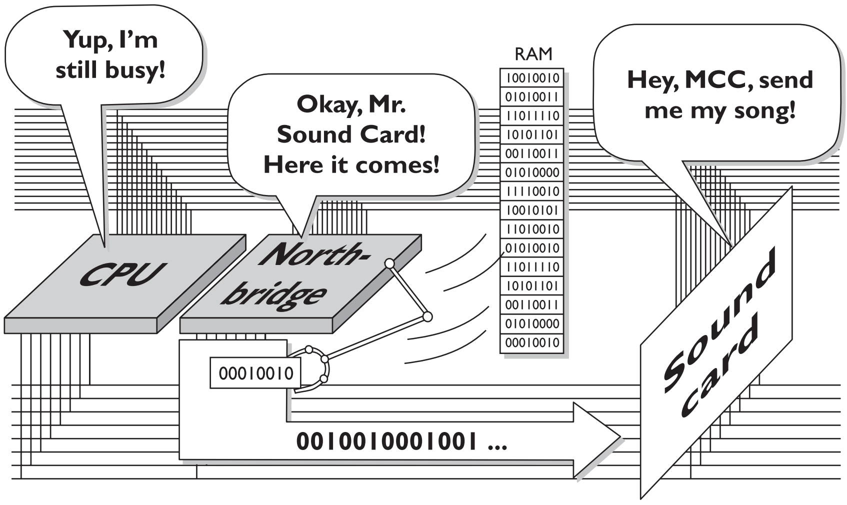


Figure 8-24 DMA in action

The *DMA controller*, which seasoned techs often call the *8237* after its old chip name, controls all DMA functions. DMA is similar to IRQ handling in that the DMA controller assigns numbers, called DMA channels, by which devices can request use of the DMA. The DMA also handles the data passing from peripherals to RAM and vice versa. This takes necessary but simple work away from the CPU so the CPU can spend time doing more productive work.

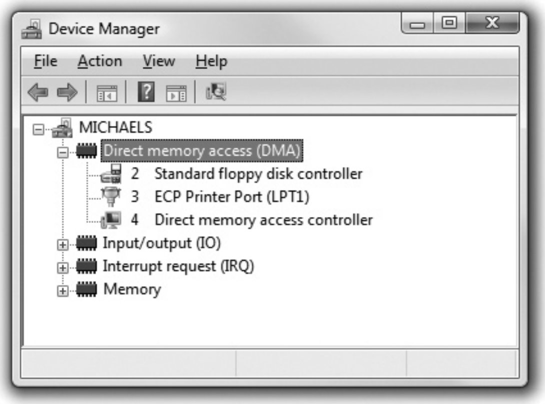
The DMA chip sends data along the external data bus when the CPU is busy with internal calculations and not using the external data bus. This is perfectly acceptable, because the CPU accesses the external data bus only about five percent of the time on a modern CPU.

The DMA just described is called classic DMA; it was the first and for a long time the only way to do DMA. Classic DMA is dying out because it's very slow and only supports 16-bit data transfers, a silly waste in a world of much wider buses. On most systems, only floppy drives still use classic DMA.

All systems still support classic DMA, but most devices today that use DMA do so without going through the DMA controller. These devices are known as bus masters. *Bus mastering* devices have circuitry that enables them to watch for other devices accessing the external data bus; they can detect a potential conflict and get out of the way on their own. Bus mastering has become extremely popular in hard drives. All modern hard drives take advantage of bus mastering. Hard drive bus mastering is hidden under terms such as *Ultra DMA*, and for the most part is totally automatic and invisible. See Chapter 12, "Implementing Hard Drives," for more details on bus mastering hard drives.

NOTE Bus mastering devices ignore the DMA controller; they don't have DMA channels.

If you want to see your DMA usage, head back to the Device Manager and change the view to *Resources by type*. Click on *Direct memory access* (*DMA*) and you'll see something like Figure 8-25. This system has only two DMA channels: one for the floppy drive and one for the connection to the CPU.



```
MICHAELS
  Direct memory access (DMA)
    2  Standard floppy disk controller
    3  ECP Printer Port (LPT1)
    4  Direct memory access controller
  Input/output (IO)
  Interrupt request (IRQ)
  Memory
```

Figure 8-25 DMA settings in the Device Manager

One interesting note to DMA is that neither PCI nor PCIe supports DMA, so you'll never find a DMA device that snaps into these expansion buses. A hard drive, floppy drive, or any other device that still wants to use DMA must do so through onboard connections. Sure, you can find hard drive and floppy drive cards, but they're not using DMA.

Memory Addresses

Some expansion cards need memory addresses, just like the system RAM. There are two reasons a card may need memory addresses. First, a card may have onboard RAM that the CPU needs to address. Second, a few cards come with an onboard ROM, the so-called adapter or option ROM you read about in Chapter 5, "Microprocessors." In either of these situations, the RAM or ROM must steal memory addresses away from the main system RAM to enable the CPU to access the RAM or ROM. This process is called *memory addressing*. You can see memory addresses assigned to expansion cards by clicking on Memory in the Device Manager when viewing resources by type.

The key fact for techs is that, just like I/O addresses, IRQs, and DMA channels, memory addressing is fully automatic and no longer an issue.

Installing Expansion Cards

Installing an expansion card successfully—another one of those bread-and-butter tasks for the PC tech—requires at least four steps. First, you need to know that the card works with your system and your operating system. Second, you have to insert the card in an expansion slot properly and without damaging that card or the motherboard. Third, you need to provide drivers for the operating system—that's *proper* drivers for the *specific* OS. Fourth, you should always verify that the card functions properly before you walk away from the PC.

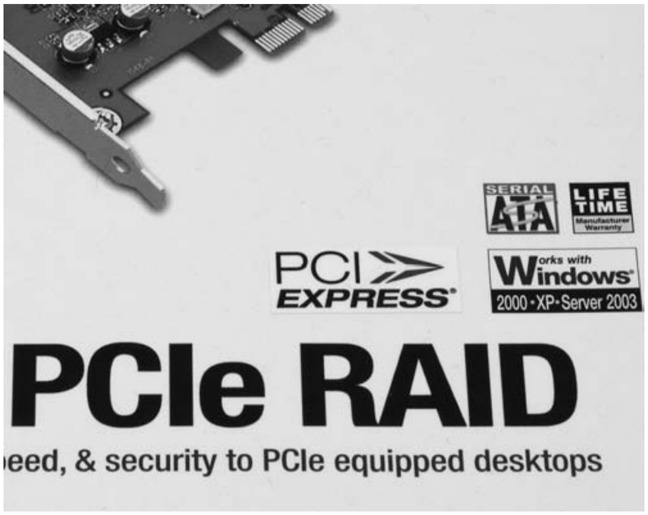
NOTE Some manufacturers insist on a different order for device installation than the traditional one listed here. The most common variation requires you to install the drivers and support software for an expansion card *before* you insert the card. Failure to follow the manufacturer's directions with such a card can lead to hours of frustration while you uninstall the card and reinstall the drivers, sometimes manually removing some drivers and software from the system. The bottom line? Read the instructions that come with a particular card! I'll provide more specific examples of problem devices in later chapters.

Step 1: Knowledge

Learn about the device you plan to install—preferably before you purchase it! Does the device work with your system and operating system? Does it have drivers for your operating system? If you use Windows, the answer to these questions is almost always "yes." If you use an old operating system such as Windows 98 or a less common operating system such as Linux, these questions become critical. A lot of older, pre-XP hardware simply won't work with Windows XP or Vista at all. Check the device's documentation and check the device manufacturer's Web site to verify that you have the correct drivers. While you're checking, make sure you have the latest version of the driver; most devices get driver updates more often than the weather changes in Texas.

For Windows systems, your best resource for this knowledge is the Windows Logo'd Products List. This used to be called the Hardware Compatibility List (HCL), and you'll still hear lots of people refer to it as such. You can check out the Web site (http://winqual .microsoft.com/hcl/Default.aspx) to see if your product is listed, but most people just look on the box of the device in question (Figure 8-26)—all Windows-certified devices proudly display that they work with Windows.

Figure 8-26 Works with Windows!


SERIAL ATA
LIFE TIME Manufacturer Warranty
PCI EXPRESS
Works with Windows 2000 • XP • Server 2003
PCIe RAID
eed, & security to PCIe equipped desktops

NOTE The Windows Vista Compatibility Center (www.microsoft.com/ windows/compatibility/) is also a great resource to check whether a particular software program works with Windows Vista.

Microsoft keeps the Logo'd Product List available for all supported operating systems, so you'll see Windows 7, Windows Vista, and most likely Windows XP (depending on when you're reading this book). Windows 2000 is already gone.

Step 2: Physical Installation

To install an expansion card successfully, you need to take steps to avoid damaging the card, the motherboard, or both. This means knowing how to handle a card and avoiding electrostatic discharge (ESD) or any other electrical issue. You also need to place the card firmly and completely into an available expansion slot.

Optimally, a card should always be in one of two places: in a computer or in an anti-static bag. When inserting or removing a card, be careful to hold the card only by its edges. Do not hold the card by the slot connectors or touch any components on the board (Figure 8-27).

Use an anti-static wrist strap if possible, properly attached to the PC, as noted in Chapter 2, "Operational Procedures." If you don't have a wrist strap, you can use the tech way of avoiding ESD by touching the power supply after you remove the expansion

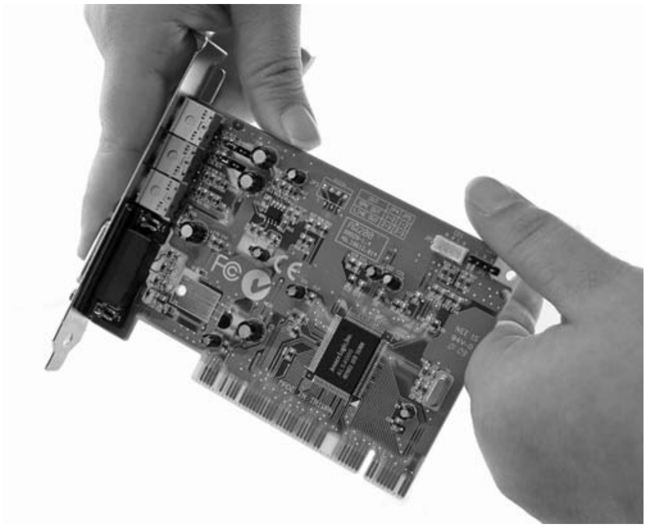


Figure 8-27 Where to handle a card

card from its anti-static bag. This puts you, the card, and the PC at the same electrical potential and thus minimizes the risk of ESD.

Modern systems have a trickle of voltage on the motherboard at all times when the computer is plugged into a power outlet. Chapter 10, "Power Supplies," covers power for the PC and how to deal with it in detail, but here's the short version: *Always unplug the PC before inserting an expansion card!* Failure to do so can destroy the card, the motherboard, or both. It's not worth the risk.

Never insert or remove a card at an extreme angle. This may damage the card. A slight angle is acceptable and even necessary when removing a card. Always screw the card to the case with a connection screw. This keeps the card from slipping out and potentially shorting against other cards. Also, many cards use the screw connection to ground the card to the case (Figure 8-28).

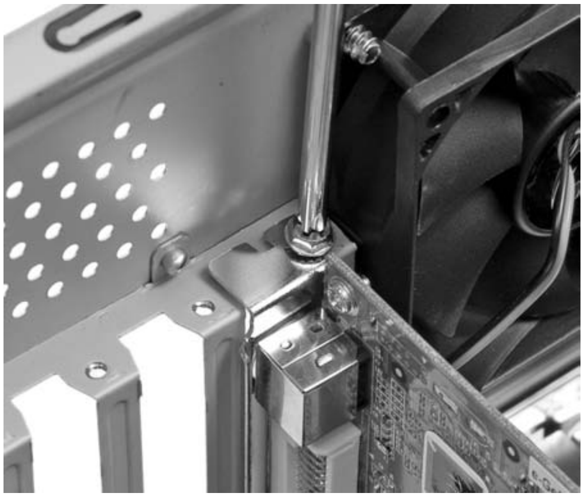


Figure 8-28 Always screw down all cards.

Many technicians have been told to clean the slot connectors if a particular card is not working. This is almost never necessary after a card is installed, and if done improperly, can cause damage. You should clean slot connectors only if you have a card that's been on the shelf for a while and the contacts are obviously dull. *Never use a pencil eraser for this purpose*. Pencil erasers can leave behind bits of residue that wedge between the card and slot, preventing contact and causing the card to fail. Grab a can of contact cleaning solution and use it instead. Contact cleaning solution is designed exactly for this purpose, cleans contacts nicely, and doesn't leave any residue. You can find contact cleaning solution at any electronics store.

A fully inserted expansion card sits flush against the back of the PC case—assuming the motherboard is mounted properly, of course—with no gap between the mounting bracket on the card and the screw hole on the case. If the card is properly seated, no contacts are exposed above the slot. Figure 8-29 shows a properly seated (meaning fitted snugly in the slot) expansion card.

Step 3: Device Drivers

You know from Chapter 7, "BIOS and CMOS," that all devices, whether built into the motherboard or added along the way, require BIOS. For almost all expansion cards, that BIOS comes in the form of *device drivers*—software support programs—loaded from a CD-ROM disc provided by the card manufacturer.

Installing device drivers is fairly straightforward. You should use the correct drivers—kind of obvious, but you'd be surprised how many techs mess this up—and, if you're upgrading, you might have to unload current drivers before loading new drivers. Finally, if you have a problem, you may need to uninstall the drivers you just loaded or, with Windows XP or Vista, roll back to earlier, more stable drivers.

Getting the Correct Drivers

To be sure you have the best possible driver you can get for your device, you should always check the manufacturer's Web site. The drivers that come with a device may work well, but odds are good that you'll find a newer and better driver on the Web site. How do you know that the drivers on the Web site are newer? First, take the easy route: look on the CD. Often the version is printed right on the CD itself. If it's not printed there, you're going to have to load the CD in your CD-ROM drive and poke around. Many driver discs have an AutoRun screen that advertises the version. If nothing is on the pop-up screen, look for a Readme file (Figure 8-30).

Driver or Device?

In almost all cases, you should install the device driver after you install the device. Without the device installed, the driver installation will not see the device and will give an error screen. The only exception to this rule is USB and FireWire devices—with these you should always install the driver first.

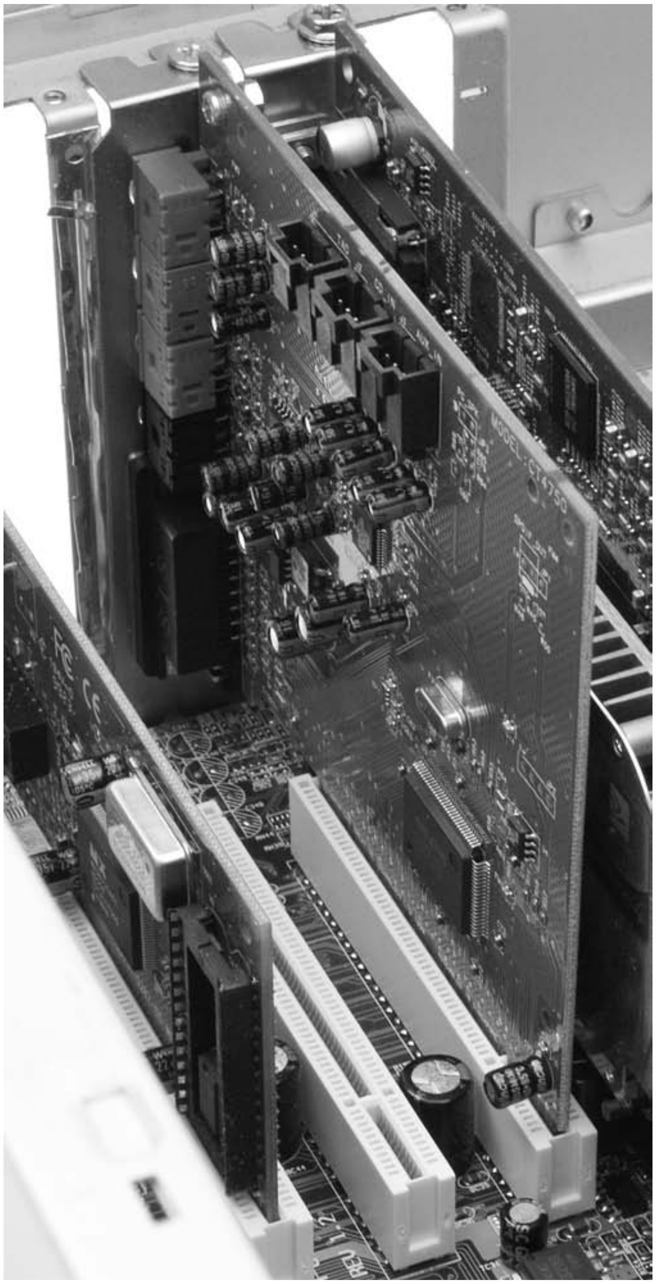
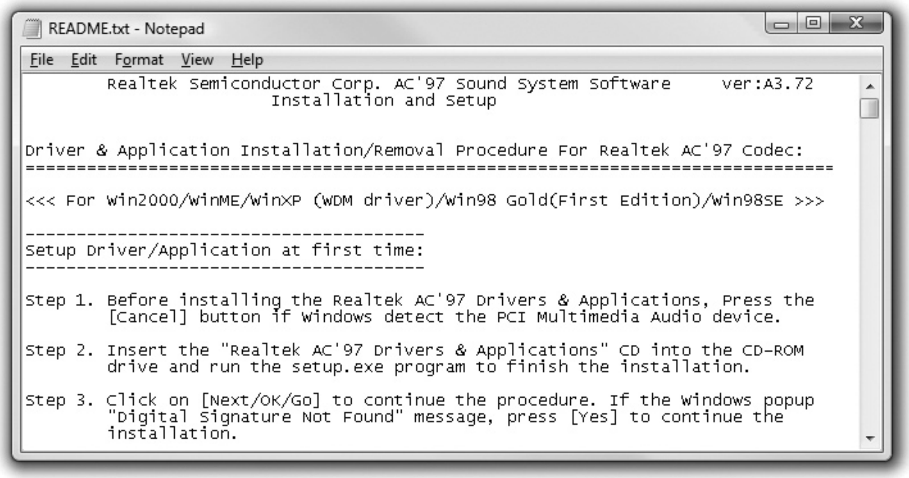


Figure 8-29 Properly seated expansion card; note the tight fit between case and mounting bracket and the evenness of the card in the slot.



```
        Realtek Semiconductor Corp. AC'97 Sound System Software     ver:A3.72
                        Installation and Setup


Driver & Application Installation/Removal Procedure For Realtek AC'97 Codec:
==========================================================================

<<< For Win2000/WinME/WinXP (WDM driver)/Win98 Gold(First Edition)/Win98SE >>>

---------------------------------------
Setup Driver/Application at first time:
---------------------------------------

Step 1. Before installing the Realtek AC'97 Drivers & Applications, Press the
        [Cancel] button if Windows detect the PCI Multimedia Audio device.

Step 2. Insert the "Realtek AC'97 Drivers & Applications" CD into the CD-ROM
        drive and run the setup.exe program to finish the installation.

Step 3. Click on [Next/OK/Go] to continue the procedure. If the Windows popup
        "Digital Signature Not Found" message, press [Yes] to continue the
        installation.
```

Figure 8-30 Part of a Readme file showing the driver version

Removing the Old Drivers

Some cards—and this is especially true with video cards—require you to remove old drivers of the same type before you install the new device. To do this, you must first locate the driver in the Device Manager. Right-click the device driver you want to uninstall and select Uninstall (Figure 8-31). Many devices, especially ones that come with a lot of applications, will have an uninstall option in the Add/Remove Programs (Windows 2000), Add or Remove Programs (Windows XP), or Programs and Features (Windows Vista) applet in the Control Panel (Figure 8-32).

Unsigned Drivers

Microsoft wants your computer to work, truly, and the company provides an excellent and rigorous testing program for hardware manufacturers called the *Microsoft Windows Logo Program*. Developers initially use software to test their devices and, when they're ready, submit the device to the *Windows Hardware Quality Labs* (*WHQL*) for further testing. Hardware and drivers that survive the WHQL and other processes get to wear the *Designed for Windows* logo. The drivers get a digital signature that says Microsoft tested them and found all was well.

Not all driver makers go through the rather involved process of the WHQL and other steps in the Windows Logo Program, so their software does not get a digital signature from Microsoft. When Windows runs into such a driver, it brings up a scary-looking screen (Figure 8-33) that says you're about to install an *unsigned driver*.

The fact that a company refuses to use the Windows Logo Program doesn't mean its drivers are bad—it simply means they haven't gone through Microsoft's exhaustive quality-assurance certification procedure. If I run into this, I usually check the driver's version to make sure I'm not installing something outdated, and then I just take my

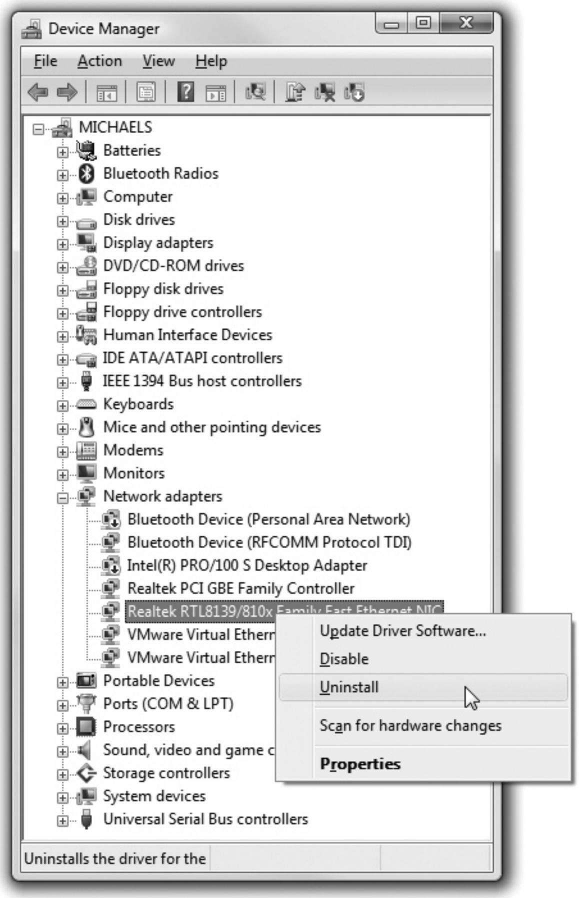


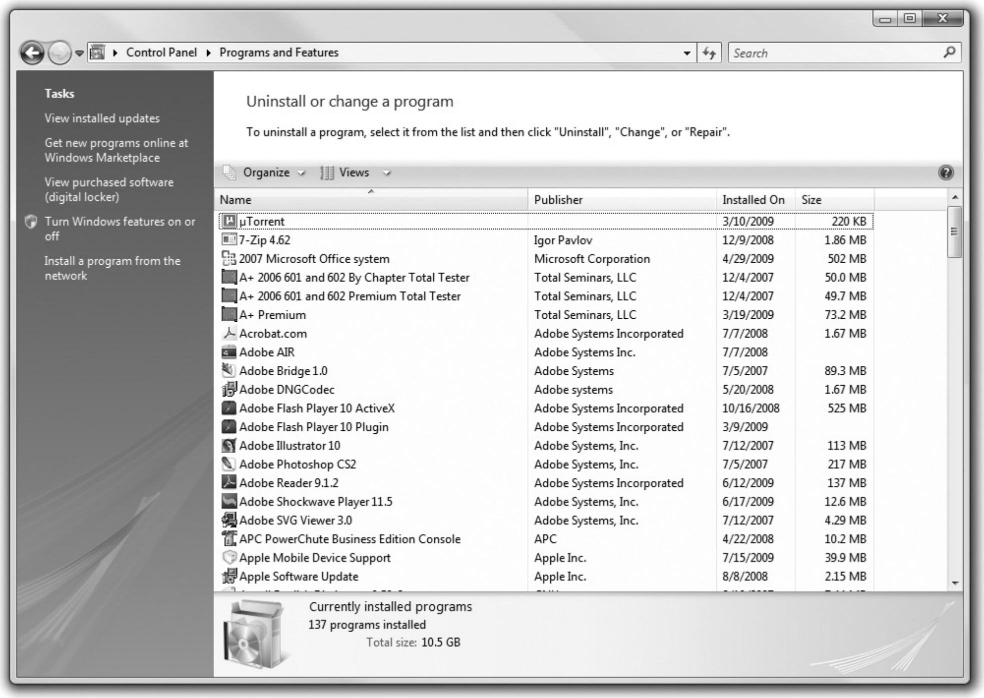
Figure 8-31 Uninstalling a device

chances and install it. (I've yet to encounter a problem with an unsigned driver that I haven't also seen with Designed for Windows drivers.)

With Windows Vista 64-bit, Microsoft tightened the rules to try to provide the most stable platform possible. You simply cannot install unsigned drivers. Microsoft must approve each one.

Installing the New Driver

You have two ways to install a new driver: by using the installation CD directly or by using the Add Hardware Wizard in the Control Panel. Most experienced techs prefer to run from the installation CD. Most devices come with extra programs. My motherboard comes with a number of handy applications for monitoring temperature and overclocking. The Add Hardware Wizard does not install anything but the drivers. Granted, some techs find this a blessing because they don't want all of the extra junk that sometimes



Control Panel ▸ Programs and Features
Search
Tasks
View installed updates
Get new programs online at Windows Marketplace
View purchased software (digital locker)
Turn Windows features on or off
Install a program from the network
Organize
Views

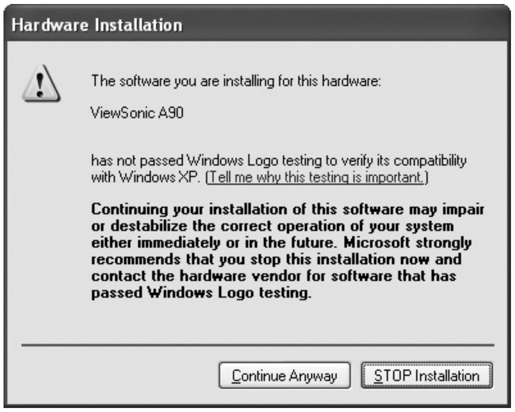
Uninstall or change a program

To uninstall a program, select it from the list and then click "Uninstall", "Change", or "Repair".

Name	Publisher	Installed On	Size
µTorrent		3/10/2009	220 KB
7-Zip 4.62	Igor Pavlov	12/9/2008	1.86 MB
2007 Microsoft Office system	Microsoft Corporation	4/29/2009	502 MB
A+ 2006 601 and 602 By Chapter Total Tester	Total Seminars, LLC	12/4/2007	50.0 MB
A+ 2006 601 and 602 Premium Total Tester	Total Seminars, LLC	12/4/2007	49.7 MB
A+ Premium	Total Seminars, LLC	3/19/2009	73.2 MB
Acrobat.com	Adobe Systems Incorporated	7/7/2008	1.67 MB
Adobe AIR	Adobe Systems Inc.	7/7/2008	
Adobe Bridge 1.0	Adobe Systems	7/5/2007	89.3 MB
Adobe DNGCodec	Adobe systems	5/20/2008	1.67 MB
Adobe Flash Player 10 ActiveX	Adobe Systems Incorporated	10/16/2008	525 MB
Adobe Flash Player 10 Plugin	Adobe Systems Incorporated	3/9/2009	
Adobe Illustrator 10	Adobe Systems, Inc.	7/12/2007	113 MB
Adobe Photoshop CS2	Adobe Systems, Inc.	7/5/2007	217 MB
Adobe Reader 9.1.2	Adobe Systems Incorporated	6/12/2009	137 MB
Adobe Shockwave Player 11.5	Adobe Systems, Inc.	6/17/2009	12.6 MB
Adobe SVG Viewer 3.0	Adobe Systems, Inc.	7/12/2007	4.29 MB
APC PowerChute Business Edition Console	APC	4/22/2008	10.2 MB
Apple Mobile Device Support	Apple Inc.	7/15/2009	39.9 MB
Apple Software Update	Apple Inc.	8/8/2008	2.15 MB

Currently installed programs
137 programs installed
Total size: 10.5 GB

Figure 8-32 The Change/Remove option in Add or Remove Programs



Hardware Installation

The software you are installing for this hardware:

ViewSonic A90

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Figure 8-33 Unsigned driver warning

comes with a device, but most installation discs give clear options so you can pick and choose what you want to install (Figure 8-34).

The other reason to use installation CDs instead of the Add Hardware Wizard stems from the fact that many expansion cards are actually many devices in one, and each device needs its own drivers. Sound cards often come with joystick ports, for example,

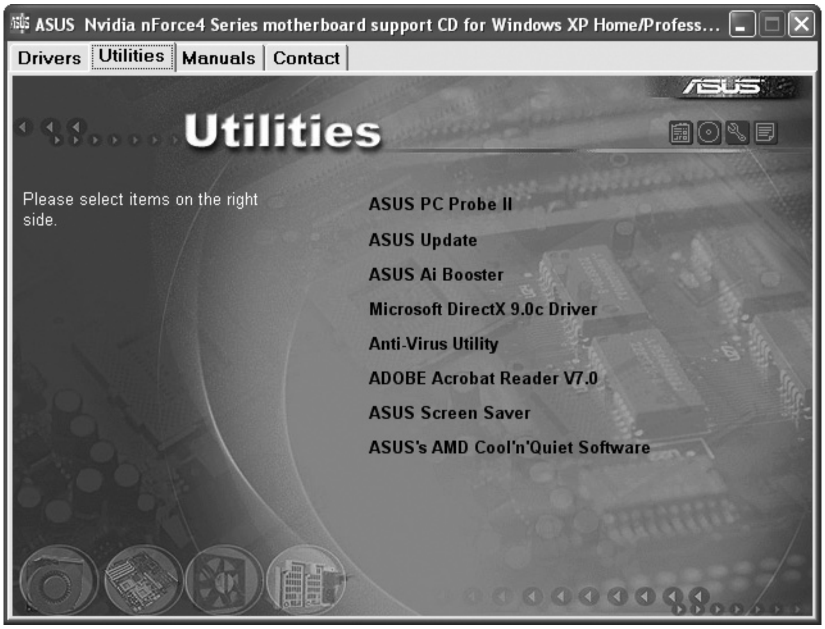

ASUS Nvidia nForce4 Series motherboard support CD for Windows XP Home/Profess...
Drivers
Utilities
Manuals
Contact
Utilities
Please select items on the right side.
ASUS PC Probe II
ASUS Update
ASUS Ai Booster
Microsoft DirectX 9.0c Driver
Anti-Virus Utility
ADOBE Acrobat Reader V7.0
ASUS Screen Saver
ASUS's AMD Cool'n'Quiet Software

Figure 8-34 Installation menu

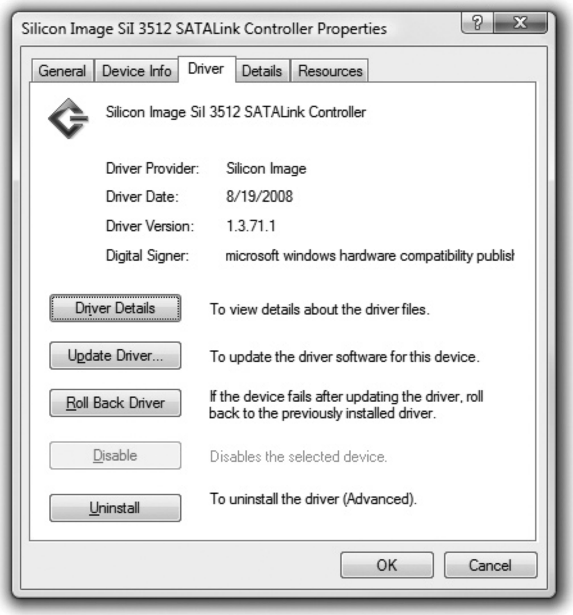
and video cards often have built-in TV tuners. The Add Hardware Wizard will install all of the devices, but the installation CD brings them to your attention. Go for the CD program first and save the Add Hardware Wizard for problems, as you'll see in the next section.

NOTE To install drivers in a Windows computer, you need to have the proper permission. I'm not talking about asking somebody if you're allowed to install the device. *Permissions* are granted in Windows to enable people to do certain things, such as add a printer to a computer or install software, or to stop people from being able to do such tasks. Specifically, you need *administrative* permissions to install drivers. Chapter 16, "Securing Windows Resources," goes into a lot of detail about permissions, so no need to worry about them here.

Driver Rollback

Windows XP and Windows Vista offer the nifty feature of rolling back to previous drivers after an installation or driver upgrade. If you decide to live on the edge and install beta drivers for your video card, for example, and your system becomes frightfully unstable, you can back up to the drivers that worked before. (Not that I've ever had to use that feature, of course.) To access the rollback feature, simply open the Device Manager and access the properties for the device you want to adjust. On the Driver tab (Figure 8-35), you'll find the Roll Back Driver button.

Figure 8-35 Driver rollback feature


Silicon Image SiI 3512 SATALink Controller Properties

Silicon Image Sil 3512 SATALink Controller

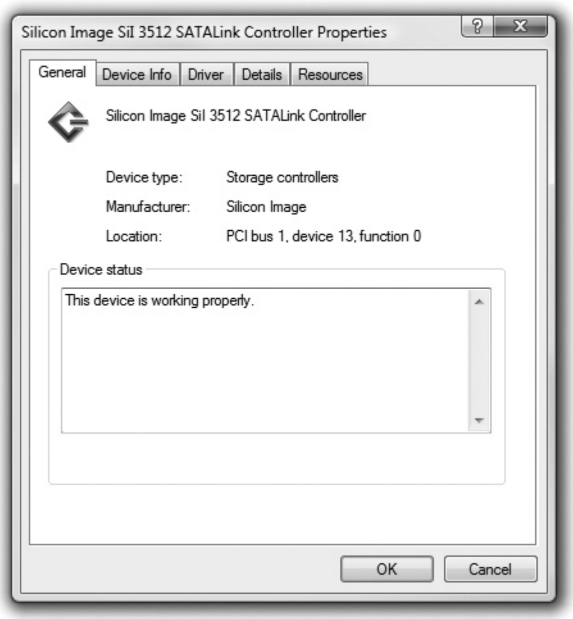
Driver Provider: Silicon Image
Driver Date: 8/19/2008
Driver Version: 1.3.71.1
Digital Signer: microsoft windows hardware compatibility publisl

Driver Details — To view details about the driver files.
Update Driver... — To update the driver software for this device.
Roll Back Driver — If the device fails after updating the driver, roll back to the previously installed driver.
Disable — Disables the selected device.
Uninstall — To uninstall the driver (Advanced).

NOTE Many PC enthusiasts try to squeeze every bit of performance out of their PC components, much as auto enthusiasts tinker with engine tunings to get a little extra horsepower out of their engines. Expansion card manufacturers love enthusiasts, who often act as free testers for their unpolished drivers, known as *beta drivers*. Beta drivers are fine for the most part, but they can sometimes cause amazing system instability—never a good thing! If you use beta drivers, make sure you know how to uninstall or roll back to previous drivers.

Step 4: Verify

As a last step in the installation process, inspect the results of the installation and verify that the device works properly. Immediately after installing, you should open the Device Manager and verify that Windows sees the device (Figure 8-36). Assuming that the Device Manager shows the device working properly, your next check is to put the device to work by making it do whatever it is supposed to do. If you installed a printer, print something; if you installed a scanner, scan something. If it works, you're finished!


Silicon Image SiI 3512 SATALink Controller Properties

General | Device Info | Driver | Details | Resources

Silicon Image SiI 3512 SATALink Controller

Device type:	Storage controllers
Manufacturer:	Silicon Image
Location:	PCI bus 1, device 13, function 0

Device status

This device is working properly.

Figure 8-36 Device Manager shows the device working properly.

Practical Application

Troubleshooting Expansion Cards

A properly installed expansion card rarely makes trouble; it's the botched installations that produce headaches. Chances are high that you'll have to troubleshoot an expansion card installation at some point, usually from an installation you botched personally.

The first sign of an improperly installed card usually shows up the moment you first try to get that card to do whatever it's supposed to do and it doesn't do it. When this happens, your primary troubleshooting process is a reinstallation—after checking in with the Device Manager.

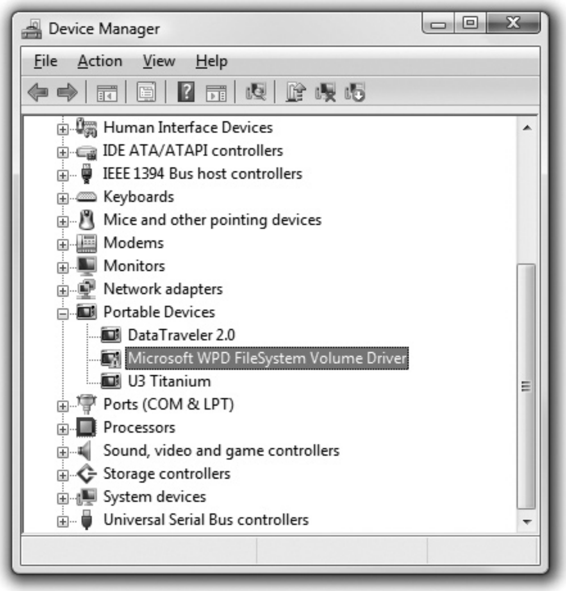
Other chapters in this book cover specific hardware troubleshooting: sound cards in Chapter 20, "Multimedia," for example, and video cards in Chapter 19, "Video." Use this section to help you decide what to look for and how to deal with the problem.

The Device Manager provides the first diagnostic and troubleshooting tool in Windows. After you install a new device, the Device Manager gives you many clues if something has gone wrong.

Occasionally, the Device Manager may not even show the new device. If that happens, verify that you inserted the device properly and, if needed, that the device has power. Run the Add/Remove Hardware Wizard and see if Windows recognizes the device. If the Device Manager doesn't recognize the device at this point, you have one of two problems: either the device is physically damaged and you must replace it, or the device is an onboard device, not a card, and is turned off in CMOS.

The Device Manager rarely completely fails to see a device. More commonly, device problems manifest themselves in the Device Manager via error icons—a black "!" or a red "X" or a blue "i."

- A black "!" on a yellow circle indicates that a device is missing (Figure 8-37), that Windows does not recognize a device, or that there's a device driver problem. A device may still work even while producing this error.

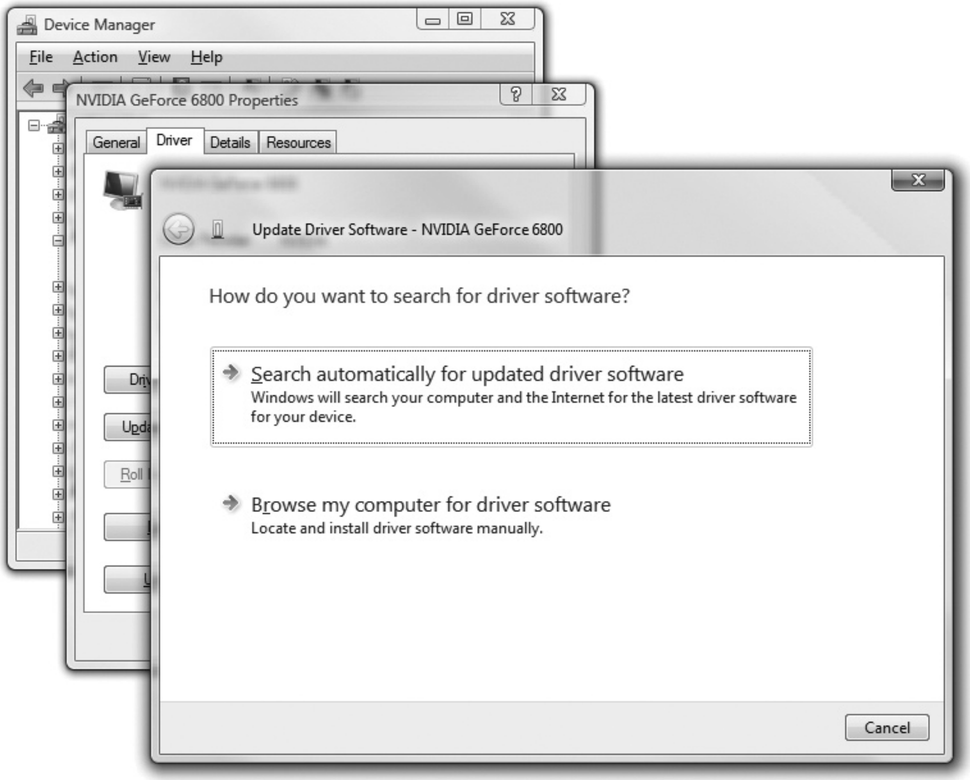


Human Interface Devices
IDE ATA/ATAPI controllers
IEEE 1394 Bus host controllers
Keyboards
Mice and other pointing devices
Modems
Monitors
Network adapters
Portable Devices
DataTraveler 2.0
Microsoft WPD FileSystem Volume Driver
U3 Titanium
Ports (COM & LPT)
Processors
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

Figure 8-37 An "!" in the Device Manager, indicating a problem with the selected device

- A red "X" indicates a disabled device. This usually points to a device that's been manually turned off, or a damaged device. A device producing this error will not work.
- A blue "i" on a white field indicates a device on which someone has configured the system resources manually. This only occurs on non-ACPI systems. This symbol merely provides information and does not indicate an error with the device.

The "!" symbol is the most common error symbol and usually the easiest to fix. First, double-check the device's connections. Second, try reinstalling the driver with the Update Driver button. To get to the Update Driver button, right-click the desired device in the Device Manager and select Properties. In the Properties dialog box, select the Driver tab. On the Driver tab, click the Update Driver button to open the updating wizard (Figure 8-38).


Device Manager
File Action View Help
NVIDIA GeForce 6800 Properties
General Driver Details Resources
Update Driver Software - NVIDIA GeForce 6800
Cancel

How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device.

Browse my computer for driver software
Locate and install driver software manually.

Figure 8-38 Updating the driver

A red "X" error strikes fear into most technicians. If you get one, first check that the device isn't disabled. Right-click on the device and select Enable. If that doesn't work (it often does not), try rolling back the driver (if you updated the driver) or uninstalling (if it's a new install). Shut the system down and make triple-sure you have the card physically installed. Then redo the entire driver installation procedure, making sure you have the most current driver for that device. If none of these procedures works, return the card—it's almost certainly bad.

As you look at the errors in the Device Manager, you'll notice error codes for the device that does not work properly. Windows has about 20 error codes, but the fixes still boil down to the same methods just shown. If you really want to frustrate yourself, try the Troubleshooter. It starts most fixes the same way: by reinstalling the device driver.

Chapter Review Questions

1. Which of the following statements about the expansion bus is true?
   - A. The expansion bus runs at the speed of the system clock.
   - B. The expansion bus crystal sets the speed for the expansion bus.
   - C. The CPU communicates with RAM via the expansion bus.
   - D. The frontside bus is another name for the expansion bus.
2. Which of the following statements best describes the hexadecimal numbering system?
   - A. It is a binary numbering system that uses only two digits, a zero and a one.
   - B. It is another name for the decimal numbering system with ten digits, zero through nine.
   - C. It is an eight-digit numbering system, using letters A through H.
   - D. It is a sixteen-digit numbering system representing binary values, using the characters zero through nine and A–F.
3. AGP stands for ______________ and is an expansion of the _____________ bus.
   - A. Accelerated Graphics Port, PCI
   - B. Alternative Graphics Port, PCI
   - C. Accelerated Graphics Port, ISA
   - D. Alternative Graphics Port, ISA
4. Which of these devices is likely to still use DMA?
   - A. USB flash drive
   - B. Floppy drive
   - C. Hard drive
   - D. CD-ROM drive
5. What does a red "X" next to a device in the Device Manager indicate?
   - A. A compatible driver has been installed that may not provide all of the functions for the device.
   - B. The device is missing or Windows cannot recognize it.
   - C. The system resources have been assigned manually.
   - D. The device has been disabled because it is damaged or has a system resource conflict.
6. When installing an expansion card, which of these should you do?
   - A. Make sure the computer is plugged in to the AC wall outlet.
   - B. Hold the card only by its slot connectors.

   C. Use firm but not excessive force to snap the card into the slot.
   D. Avoid letting the metal flange touch the PC case.

7. How does the CPU communicate with a device?
   A. It uses the device's I/O addresses over the address bus.
   B. It uses the device's IRQ addresses over the data bus.
   C. It uses the device's COM port over the address bus.
   D. It uses the device's DMA over the data bus.

8. Which of the following does a device use to initiate communication with the CPU?
   A. IO/MEM wire
   B. Bus mastering
   C. DMA
   D. IRQ

9. Which variation of the PCI bus was specifically designed for laptops?
   A. PCI-X
   B. PCIe
   C. Mini-PCI
   D. AGP

10. Which of the following bus types uses serial rather than parallel communication?
    A. AGP
    B. PCI
    C. PCIe
    D. PCI-X

Answers

1. **B.** A separate expansion bus crystal enables the expansion bus to run at a different speed than the frontside bus.
2. **D.** The hexadecimal numbering system represents binary values using the characters zero through nine and A–F.
3. **A.** The Accelerated Graphics Port is a specialized PCI slot used for video cards.
4. **B.** On most modern PCs, only the floppy drive (if one is installed) still uses classic DMA.
5. **B.** The dreaded red "X" can mean a bad connection, a bad driver, or even a bad card.

6. **C.** After handling the card only by the edges and avoiding the slot connectors, you should snap it into an open slot on an unplugged system, pushing firmly and evenly until the metal flange is in contact with the slot on the case.
7. **A.** The CPU uses the device's I/O addresses over the address bus to communicate with that device.
8. **D**. A device uses its IRQ to get the CPU's attention and begin communication.
9. **C.** The Mini-PCI format conserves space and power, making it an ideal card type for use in laptops.
10. **C.** The PCI Express bus uses serial rather than parallel communication.