Slot machines

I/O... I/O... it's off to work we go on PCI buses. Roger Gann looks at developments and conflicts.

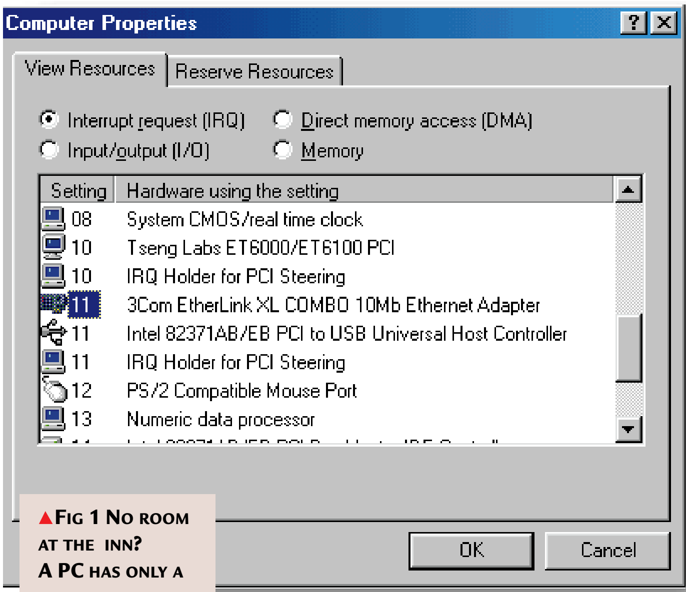
We've had the familiar PCI (Peripheral Component Interconnect) system bus for five years now, and those short, white expansion slots are now a familiar sight on all motherboards, not just those found in PCs. The PCI bus presented significant advantages over the old, 16-bit, ISA slot: with a 32-bit bus running at 33MHz, it offered twice the bandwidth and four times the speed; two things of which you can never have enough. In theory, the PCI bus could deliver 132Mbps, but in practice it topped out at around 90Mbps, the rest being taken up with overheads. On top of this, PCI offered other advantages, such as Plug-and-Play.

Just recently, servers have started to surface which sport 64-bit PCI slots running at 66MHz, further quadrupling data throughput. Yet even this kind of bandwidth will be stressed by up-and-coming technologies such as Ultra3 SCSI, Fibre Channel, FireWire and Gigabit Ethernet.

In September '98, Compaq, Hewlett-Packard and IBM announced the PCI-X bus specification and then submitted it to the PCI Special Interest Group. The new bus specification increases the PCI bus speed to 133MHz and increases the amount of data it can handle during each clock cycle to 66 bits, with a theoretical maximum data throughput of nearly 1Gb/sec. This initiative, on the part of the three server manufacturers, was a rather surprising move.

Intel was conspicuous by its absence — it was the designer of the original 33MHz and 66MHz PCI bus standards. Some observers were surprised that it took major vendors like IBM so long to wrench the initiative from Intel. The chip giant has struggled to deliver enterprise-related solutions: Intel's Xeon has been affected by scalability issues, and the much-vaunted Merced processor has been delayed until potentially 2000.

The specification proposal is aimed at addressing I/O bandwidth for servers running enterprise applications such as Gigabit Ethernet, Fibre Channel and Ultra3 SCSI. Currently, the fastest PCI bus design runs at 66MHz and can pass up to 64 bits of information per clock cycle. The bus standard found in the majority of servers and desktop computers is the slower 33MHz PCI bus with a 32-bit gate developed by Intel in 1993.



▲**Fig 1 No room at the inn? A PC has only a limited number of IRQs**

The X factor

The new PCI-X specification features a new multi-slot design that has not yet been implemented. It will support one slot running at 133MHz, two slots at 100MHz and four slots at 66MHz. This allows for greater flexibility in implementation, permitting different configurations for different applications. However, it's unlikely that even in the fastest one-slot configuration of 133MHz, the theoretical throughput of 1Gb/sec will be reached; 800Mb/sec is more achievable.

Intel has struggled to deliver enterprise-related solutions

PCI-X also features an enhanced protocol to increase the efficiency of data transfer and to simplify electrical timing requirements, which is an important factor at higher clock frequencies. Allowances for future capabilities are also provided. In addition, PCI-X is designed to accomplish this while remaining backwards-compatible with the current PCI bus. PCI-X is best used in "enterprise" storage and networking applications where there's a need to move data around the system.

PORTABLE PCI

It has been long mooted, yet at long last there seems to be movement on the PCI-for-notebooks front. Notebooks are too small to take conventional expansion cards, so users have to resort to docking stations or PC Cards whenever they need to plug something even slightly exotic into a notebook. A PCI Special Interest Group is looking at a new specification for a small-form-factor PCI solution, possibly a daughtercard, for notebooks. Apparently, the new PCI daughtercard could be implemented in a number of ways, perhaps on the side of a notebook in the form of a dongle connector, or wholly contained within a notebook.

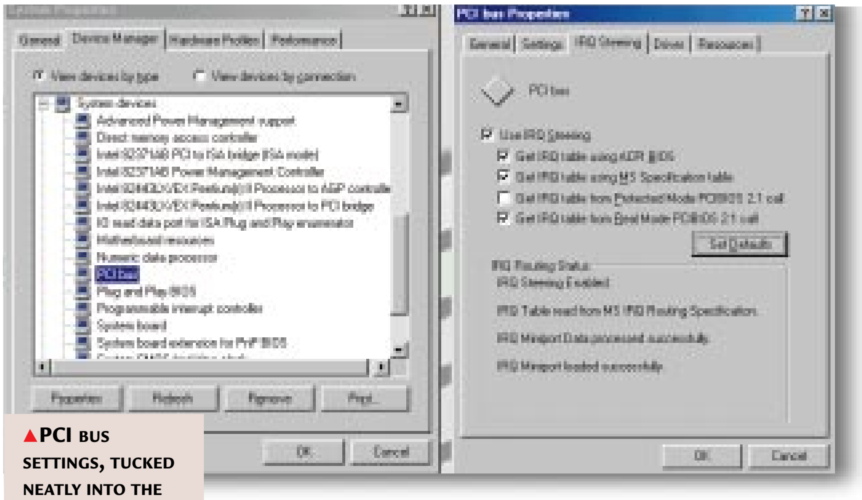
Intel responded quickly to PCI-X by announcing a future bus technology at the IDF developer conference. Called Next Generation I/O, Intel claims it will bring mainframe-class links to PCs, internet working gear and networked peripherals. Intel has described NGI/O as a fast serial bus, "unchained" from host interrupts and better characterised as a switching fabric, linking a processor complex to internal and external systems peripherals. Phew!

Resource conflicts

The most common resource conflicts involve interrupt requests, or IRQs, with ISA bus-based peripherals. IRQ conflicts have been the bane of end-users for many years but they're becoming increasingly common because more and more computers are being shipped with few, if any, available IRQs.

Many moons ago (well, June 1998, to be precise) I was whingeing in this column about the chronic lack of hardware IRQs available in current PCs. I had found to my horror that some PCs actually ship from the factory with no free IRQs, so stuffed are they with add-ons. As a result, I found it incredibly difficult to install something as simple as a network interface card, a job which should have taken me and Plug-and-Play ten minutes at the most, but took me no less than three evenings to "kind of" sort out. The problem is that there are only 16 IRQs in a PC and typically only five or fewer are available for future expansion. "Why not add more IRQs by installing additional interrupt controllers?" is one seemingly simple solution to the IRQ famine, but this fix is typically only found in exotic hardware such as high-end workstations and servers simply because of the costs and complexities involved in implementing them.

PC manufacturers have, instead, tried to resolve the problem by a different approach, namely to avoid using IRQs. PCI was supposed to help out here but PCI devices often do use IRQs, sometimes multiple IRQs, which makes even fewer available for other devices



▲**PCI BUS SETTINGS, TUCKED NEATLY INTO THE SYSTEM PROPERTIES DIALOGUE**

such as Plug-and-Play-compatible ISA cards. Resolving IRQ problems, far from getting simpler, has actually become harder.

Where you're using ISA and PCI cards together, it appears that PCI devices generally have to use at least one IRQ to maintain compatibility with today's operating systems. Perhaps in the future, when the ISA bus has disappeared courtesy of the PC 99 specification, PCI IRQs won't be an issue, but for the moment they are.

So what about PCI-sharing (aka PCI Steering)? Multiple ISA cards can't share an ISA IRQ, but multiple PCI devices can share a PCI IRQ. On computers that use a PCI bus, the 16 standard IRQs can be programmed to either PCI or ISA mode. An IRQ cannot be programmed for both modes at once. The ability for several PCI devices to share that single IRQ is largely determined by the computer's BIOS. Some BIOSes have settings that let you enable PCI IRQ sharing, but many do not. If the BIOS doesn't support sharing, PCI devices, including the PCI-to-Universal Serial Bus controller found in most new machines, will each use an IRQ. There are four shareable IRQs assignable to PCI devices, so if you fit four PCI cards, each will get its own unique IRQ. Add a fifth PCI card and it would automatically share with one of the first four devices. Two other PCI IRQs (15 and 16) are typically dual-PCI

You can take an IRQ horse to water but you can't make it share

Integrated Device Electronics (IDE) controllers found on most machines, but are not shared for reasons of backwards compatibility.

In theory you can, if you have Windows 95 OSR2.n or Windows 98, solve this problem by turning on an option called IRQ Steering. This was a little flakey under OSR2 but the Windows 98 implementation seems much better. By using PCI-bus IRQ steering, Windows can dynamically assign or "steer" PCI-bus IRQs to PCI devices. Note that the original releases of Windows 95 and Microsoft Windows 95 OSR1 don't provide support for PCI-bus IRQ steering. In these releases, the BIOS assigns IRQs to PCI devices.

With OSR2, if PCI-bus IRQ steering is disabled in Windows, the BIOS assigns IRQs to PCI devices. But if PCI-bus IRQ steering is *enabled*, Windows assigns IRQs to PCI devices. In the latter case, the BIOS still assigns IRQs to PCI devices, and even though Windows has the ability to change these settings, it generally doesn't.

PCI-bus IRQ steering gives Windows 95 OSR2 and later the flexibility to reprogram PCI interrupts when "rebalancing" Plug-and-Play PCI and ISA resources around non-Plug-and-Play ISA devices. In Fig 1 [p277] you can see that my network interface card is using IRQ 11, as is the USB controller. Both are PCI devices, so the IRQ they're using (IRQ 11) can be shared.

Note that there's an IRQ Holder for PCI Steering sitting on that IRQ as well. An IRQ Holder for PCI Steering indicates that an IRQ has been programmed to PCI mode and is unavailable for ISA

THE NEW ULTRA DMA/66 INTERFACE

Hard on the heels of a faster PCI bus comes a faster hard disk. About now, hard-disk drives sporting the new Ultra DMA/66 interface will begin to appear. Also known as Ultra ATA/66, this new interface is a low-cost extension of the previous Ultra DMA/33 hard-drive interface that doubles its burst data transfer rate to 66.6 megabytes per second; twice that of Ultra DMA/33's 33.3Mb/sec. Hard-disk data transfer rates have been doubling every few years: in 1994 we had DMA Mode 2/PIO Mode 4 running at 16.6Mb/sec, and in 1997 we had Ultra DMA/33 at 33.3Mb/sec. In 1999, it's time to double it yet again.

Ultra ATA/66 also delivers improved data integrity to the EIDE interface through the use of a new-style 40-pin 80-conductor cable and Cyclic Redundancy Check (CRC) error detection code. The new cable still has 40 pins, but each data cable is now shielded by its own "ground cable" which reduces crosstalk and improves signal integrity.

The new Ultra DMA/66 connector remains plug-compatible with existing 40-pin headers and the extra cost for the cable should be minimal. Ultra ATA/66 hard drives will be backwards-compatible with Ultra ATA/33 and DMA, and with existing EIDE/IDE hard drives, CD-ROM drives and host systems.

As hard-disk capacities mushroom and rotational speeds continue to accelerate, so the drive's internal transfer rate increases. The transfer of large files, often written sequentially on the hard drive, is particularly affected by the transfer rate. During sequential reads, the hard drive, because of its fast internal data rate, may fill its buffer faster than the host can empty it when using the Ultra ATA/33 or the older multi-word DMA interfaces. Improving the interface to keep up with internal data rate improvements is just what Ultra DMA/66 can achieve.

Real-world gains delivered by Ultra DMA/66 won't be earth-shattering

Although 66.6Mb/sec sounds a lot, the day-to-day real-world gains delivered by Ultra DMA/66 won't be earth-shattering (the 66.6Mb/sec figure is a burst-mode value): sustained data transfer rates will be but a fraction of this. Its benefit will be most noticeable during long sequential operations such as replaying audio or movie clips. To use Ultra ATA/66 technology, a PC must have:

- Ultra ATA/66 compatible logic either on the system motherboard or on an Ultra DMA PCI adapter card.
- Ultra DMA compatible BIOS.
- A DMA-aware device driver for the operating system or an Ultra ATA/66-compatible IDE device like a hard drive or CD-ROM drive 40-pin 80-conductor cable.

You'll need no special Windows 9x drivers to avail yourself of the performance benefits of Ultra DMA/66. As long as Windows 9x is currently using DMA mode transfers, then it will be Ultra DMM/33 or 66 "agnostic" when you upgrade your hardware.

	Data Transfer Rate (max)	Connector	Cable	CRC
DMA Mode 1	11.1Mb/sec	40-pin IDE	40-way	No
Multi-word DMA Mode 1	13.3Mb/sec	40-pin IDE	40-way	No
Multi-word DMA Mode 2	16.6Mb/sec	40-pin IDE	40-way	No
Ultra ATA Mode 2	33.3Mb/sec	40-pin IDE	40-way	Yes
Ultra ATA Mode 4	66.6Mb/sec	40-pin IDE	80-way	Yes

devices, even if no PCI devices are currently using the IRQ.

Windows 95 OSR2.n disables IRQ steering by default. To turn it on, open Control Panel, then click on the System applet and open the Device Manager tree. Scroll down to the System Devices item and get the properties of the PCI bus. One of the items here is IRQ Steering. IRQ Steering and the associated "Get IRQ Table from PCIBIOS 2.1 Call" checkbox allow Windows 9x to help manage the assignment of PCI devices to IRQs, taking over some of this capability from the BIOS. As you can see from Fig 2 [p278], IRQ steering is enabled and working OK. Sometimes, as an aid to troubleshooting conflicts, you may have to progressively uncheck these options.

IRQ steering is a godsend but don't forget you can take an IRQ horse to water but you can't make it share. In some cases, although they ought to, some PCI devices simply refuse to share IRQs.

Finally, here's one last-resort trick that's worth a spin: try moving a PCI card to a different PCI slot. On many PCs, the internal PCI Interrupts, commonly labelled A-D, are actually assigned to individual slots instead of individual cards. Often you can specify which slot gets which IRQ in CMOS setup. Because the PCI Interrupts are each commonly mapped to different IRQs, making this move can change the IRQ that a PCI card uses.

PCW CONTACTS

Roger Gann *can be contacted via the PCW editorial office (address, p10) or email* **hardware@pcw.co.uk**