Re: IRQ assignment in Windows 2K/XP/2003...

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.development.device.drivers/2005−04/msg01320.h

- *From*: "Cosmo" <cosmo_matt@xxxxxxxxxxx>
- *Date*: 29 Apr 2005 12:42:50 −0700

I never did report what I found out on this:

I discovered in the PCI bridge spec (chapter 9, revision 1.1) that if multiple devices on the other side of a PCI−to−PCI bridge (on an expansion board) are required to use the same PCI slot interrupt line (INTA#−INTD#) and therefore same IRQ#, the "interrupt pin" request in each of the device's PCI configuration header has to be different (they both don't request INTA# as I thought). For the first device, it had to specify INTA# and the second device needed to specify INTD#. When BIOS POST code is initializing the system, it assumes a pre−defined set of routing information and writes a common IRQ# to each device's "interrupt line" register in the configuration space.

Who would have thunk it?

So, no driver, Windows PnP manager, or BIOS problems were the culprit−−just a misconfiguration in hardware...

So Mark, the bridge wasn't "directly" involved with the routing of the interrupts−−but the presence of the bridge did change the configuration requirements for the devices on the secondary bus...

Cosmo

Mark Roddy wrote:
> Paul L wrote:
> > Why do you say that the motherboard has them or'd together? The Int A lines

> > do not need to be bussed together. The mother board can route the
Int lines
> > from each slot anyway it wishes.
> >
>
> Yes indeed but from the description of the OP's problem it would
appear
> that the lines are ORd together and that the BIOS is misreporting
them
> as separate. It is not that it has to be this way, it is that this
> appears, from the evidence presented, to be a fact in this situation.
>
> He has two devices behind the bridge, each of which are given
separate
> interrupt vectors by the OS, indicating that the OS believes them to
be
> separately routable. Call them DevA and DevB. DevB asserts an
interrupt,
> but the ISR for DevA gets invoked. DevA.ISR returns False, as the
> interrupt is not for DevA. DevB.ISR does not get invoked. Instead, as

> the PCI interrupt is still asserted, DevA.ISR is immediately
re−invoked.
> Repeat ad nauseum. I say that the bios is misreporting the interrupt