

Re: [PATCH 2/2] Initial generic hypertransport interrupt support.

## Re: [PATCH 2/2] Initial generic hypertransport interrupt support.

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-07/msg04561.html>

---

- *From:* [ebiederm@xxxxxxxxxxxxx](mailto:ebiederm@xxxxxxxxxxxxx) (Eric W. Biederman)
  - *Date:* Thu, 13 Jul 2006 09:13:39 -0600
- 

Dave Olson <[olson@xxxxxxxxxxxxx](mailto:olson@xxxxxxxxxxxxx)> writes:

On Wed, 12 Jul 2006, Eric W. Biederman wrote:

| Dave Olson <[olson@xxxxxxxxxxxxx](mailto:olson@xxxxxxxxxxxxx)> writes:

|

| > On Tue, 11 Jul 2006, Eric W. Biederman wrote:

| > | There is a hypertransport capability that implements a rough equivalent  
| > | of a per device ioapic. It is quite similar to MSI but with a different  
| > | register level interface.

| >

| > It's really just the same as MSI, and is set up and handled pretty  
| > much the same way.

|

| No it is not just the same. There is not global enable bit, only

...

| But from the perspective of using them in a driver the concept really  
| is the same.

I meant, from the perspective of a driver. Sorry for not being clear.

Yes. Once you figure out which one you want the concept and get an irq using the in the driver is the same. The driver does need to be aware of the details so it knows which interrupt it is generating though.

| The code that breaks it is only in -mm. It's scheduled for 2.6.19.  
| All of the MSI magic in ioapic land on i386 and x86\_64 is deleted.  
| The code just needs to age a bit and let the few unexpected  
| corner case crop up, and get sorted out.

Well, if that set of patches is accepted into 2.6.19, it will likely break other people with HyperTransport chips and cards as well. I do know of other HTX-slot cards in development, and some of them, at least, do use interrupts.

Re: [PATCH 2/2] Initial generic hypertransport interrupt support.

And I tested it on one of them. The problem is that there is no API in the kernel for properly handling hypertransport interrupts or even faking it well currently. There is no shame in breaking a bad unmaintainable hack, as I did. The responsible thing is to when you find one to fix up the code so that things work by design in a maintainable way, which I am attempting to do.

So I think it needs to age enough to not break existing drivers.

All existing drivers that use HT interrupts are broken by design. Nor do out of tree kernel drivers count, because the authors are not working with the community. I can almost use the fact that the code breaks out of tree drivers as an argument for mainstream kernel inclusion.

Unfortunately, I still haven't had time to try your HT path with our ipath driver, because I'm in fire-fighting mode on some customer issues. You might be able to test it on some of your LS-1 systems in your lab, if you need early feedback. I'll try it, as soon as I can.

Sure. In that case can I please have a good description of what weird hacks your hardware designers have done. As I understand it I cannot write to the standard registers HT capability registers and have things work correctly.

| > This part I never really quite understood. Why do you want a separate  
| > interface than the existing request\_irq().  
|  
| request\_irq is still needed. The question is how do you get the irq.  
|  
| > and pci\_enable\_msi()?  
|  
| The HT and msi semantics are moderately different, but I have  
| implemented the equivalent of pci\_enable/disable\_msi. So the  
| code is not a pci standard but just a ht standard I didn't use the  
| pci prefix.

My point was that many other pci\_\* functions have to be called by any driver for HyperTransport-attached chips (that aren't chipset), so why break these out separately, rather than somehow fitting them into the existing routines, perhaps by looking at the bus-type the device is attached to?

Re: [PATCH 2/2] Initial generic hypertransport interrupt support.

Re: [PATCH 2/2] Initial generic hypertransport interrupt support.

Until you have time to look and think about this I'm not prepared to discuss this. All I did was add code to work with the standard ht irq capability.

Unless we add a full set of ht\_\* routines (not something I'd like to see!) I don't see a reason to do it for just these routines, other than convenience for early testing of patch proposals, as opposed to final code going into the mainline.

I'm not suggesting re-using existing (old) msi.c code. I'm suggesting making the new MSI code work for all 3 of PCI, PCIe, and HT, with the same exact exported routine.

msi and msi-x are different, so to is ht. There is no real commonality there.

| Of course I don't expect the interface exported to drivers to change.

But if you add new ht\_\* routines that they have to call, that's definitely a change, a new set of exported routines. I'm questioning whether that's really the right direction. If you had already intended to have them not exported in the final patches, that wasn't clear to me, and I apologize for misunderstanding.

The functions I exported I intend to export. The complaint seems to be that you don't have anything that will work on earlier kernels. I have to agree you don't.

Eric

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>

Re: [PATCH 2/2] Initial generic hypertransport interrupt support.