

Re: [lm-sensors] 2.6.24-rc4 hwmon it87 probe fails

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-12/msg08266.html>

- *From:* "Jean Delvare" <khali@xxxxxxxxxxxxx>
 - *Date:* Sun, 23 Dec 2007 10:28:05 +0100
-

Hi Bjorn,

Le 23/12/2007, Bjorn Helgaas a écrit:

On Saturday 22 December 2007 4:21:41 am Jean Delvare wrote:

This patch makes the it87 driver request only the two ports used for the Environment Controller device.

The problem is that the IT87xxF chips do decode 4 ports (recent chips, 0x294-0x297) or 8 ports (older chips, 0x290-0x297), not 2 as the datasheets say. The ITE Super-I/O chips differ from the Winbond Super-I/O chips in this respect. The it87 driver only needs to access ports 0x295 and 0x296, true, but the device itself decodes more addresses than that. So, with this proposed patch, ports which are busy will be shown as being free. This pretty much voids the point of resource declarations, doesn't it? This might not cause too much trouble in practice, but to me this still looks like the wrong way to go.

Yes, all the ports decoded by the chip should certainly be reserved, but I think the entire range should be reserved at a higher level, like the PNP core, and the driver should reserve only the ports it uses. Then the entire range is reserved even if there is no driver or the driver is not loaded.

The problem is that the it87 driver is used on a variety of motherboards, some where the hardware monitoring device I/O ports are reserved by the BIOS, some where they aren't. How am I supposed to deal with this?

That's the approach we use for PCI, e.g.,

e8100000-e81fffff : 0000:00:08.0 <--- reserved by PCI core

Re: [lm-sensors] 2.6.24-rc4 hwmon it87 probe fails

e810000-e8102fff : CS46xx_BA1_data0 <-- reserved by driver
e8110000-e81137ff : CS46xx_BA1_data1
e8120000-e8126fff : CS46xx_BA1_pmem
e8130000-e81300ff : CS46xx_BA1_reg

PCI is an entirely different beast. With PCI you know the PCI device ID that is associated with the resources, and for a given device, the resources are always declared (if standard BARs are used) or never declared; there's no "may be". So it's much easier to handle the resources properly.

The resource declarations made by these motherboard BIOSes are totally bogus. 0x290-0x29f is much larger than what the chip decodes. 0x290-0x294 is a subset that doesn't make any sense. The GA-K8N Ultra 9 is even funnier with 0x295-0x314, which again doesn't correspond to anything real.

I agree those declarations are probably wrong. But at least they're larger than required, so they should be safe.

That's not really safe, no. They may overlap with other device resources and prevent those drivers from loading.

All these happen to not intersect with 0x295-0x296 but I wouldn't count on it. If Gigabyte (and possibly others) care that little about these declarations, pretty much anything could be seen. So while your proposed workaround happens to fix the problem at hand, it is not really correct technically, and could break again soon.

I'd rather fix the problem at its source, or, as fixing it at the source isn't very realistic in this case, as near of the source as possible. That would mean DMI-based quirks for the affected motherboards, that would discard or adjust the bogus resource declarations.

Well, I think the driver change *is* correct, assuming that the entire range can be reserved at a higher level. In this case, it is, via a PNP0C02 device.

As I wrote above, the problem is that you can't assume that. Some motherboards do declare the range at PNP level but some don't. That's the reason why the it87 driver (and most other hwmon drivers for Super-I/O devices) do declare the I/O resource again.

Another problem is how do I connect this specific I/O port range of the

Re: [lm-sensors] 2.6.24-rc4 hwmon it87 probe fails

PNP0C02 device with the it87 driver? I am by far no PNP expert but it looks to me like this PNP device covers more than one actual device, and I/O port ranges do not have labels nor identifiers that would let me find the one that corresponds to the IT87xxF device (if it exists at all.)

I'm all for integrating the it87 driver into the PNP subsystem if it is going to solve problems, but I just don't know how it works. If you do some work in this direction, I'll be happy to help with reviews and testing.

I think a DMI-based quirk to fix the PNP0C02 resources would be a good approach, but we shouldn't do that until those resources cause some other problem.

Well these resources already do cause problems, otherwise we wouldn't be having this discussion ;)

(...)

I'm sure these boxes still have PNPBIOS, so pnpacpi=off would probably work fine. But it feels like a bigger hammer than necessary for this problem.

OK.

--

Jean Delvare

--

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/>