

Re: Linux 2.6.22-rc2

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-05/msg10427.html>

- *From:* Stephen Hemminger <shemminger@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 23 May 2007 07:58:20 -0700
-

On Tue, 22 May 2007 18:53:33 -0700 (PDT)
Linus Torvalds <torvalds@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

On Tue, 22 May 2007, Stephen Hemminger wrote:

It looks like the chip reads the wrong memory sometimes. The problem happens only on the on-board NIC's and only on this kind of motherboard.

Do you know if it happens for particular addresses? (Ie, can you tell what the physical address of the descriptor is for the errors?)

I'll look but there didn't seem to be an obvious pattern when I last looked.

For testing, I have put code in to check that the receive data actually arrived before the IRQ, it triggered on my Gigabyte 925 motherboard. It appears that DMA access is messed up.

Yes, that certainly would also explain memory corruption. Either because writes went to the wrong address, or because writes went to the right address, but because an earlier IO descriptor read had gotten corrupted, the "right address" was in fact the wrong one ;)

The reason I ask whether you have some way of telling the pattern for the physical address is that one traditional cause of DMA errors is due to broken RAM remapping setup.

As an example of that – imagine that you have 1GB of RAM in the machine,

Re: Linux 2.6.22-rc2

and realize that the memory behind the 640kB → 1MB area isn't accessible, because it's taken up by the legacy ISA region.

You have two possible outcomes: either (a) the memory is just "gone", and you lost it, or (b) there is some RAM remapping in the core chipset that makes the lost 384kB show up above the 1GB mark instead.

The same "legacy ISA" hole situation happens for the "legacy PCI" hole, which is why if you have 4GB of RAM in the machine, usually you'll see 3GB at addresses 0–3GB (roughly), and then you'll see the rest at above the 4GB mark, in order to have a nice PCI hole in the 32-bit access range.

There's also the "legacy 286" hole at the 15–16MB mark (which nobody uses any more, but chipsets still inexplicably support), and the SMM remapping.

Anyway, core chipsets generally do CPU memory accesses differently from DMA accesses from the PCI bus (at a minimum, SMM is something that only the CPU can do), so I could see a situation where the remapping was set up correctly for the CPU (and perhaps for "core chipset" devices like the integrated southbridge), but devices that do DMA from the outside get screwed over.

This board doesn't have any onboard video so that helps. I am running with 2GB of memory.

I can put a card with similar chip in an X1 slot, and there are no problems. Same driver, but different bridges, and slightly different Marvell chip.

But it might not happen for all addresses. Non-remapped stuff might work well, so if there is some way of figuring out what the bad DMA address was for an erroneous access, that might offer some clues.

This board has lots of "overclocker" friendly stuff; maybe the BIOS never really sets up the PCI bridges and clocks properly.

It's hard to set up a normal PCI-PCI bridge subtly incorrectly. But special RAM timing or remapping stuff for the host bridge – sure.

It doesn't seem like a software or driver problem. I have tried tweaking PCI registers but nothing worked in this case.

Yeah, the PCI registers that would affect things like this tend to be in the host bridge, not on the normal device.

Re: Linux 2.6.22-rc2

That said, Intel doesn't generally do the really insane things. And a lot of the old remapping stuff is simply not done any more. For example, I doubt that the 925 chipset even supports remapping the 640k-1M range any more: 384kB just isn't worth it when people talk about gigs of RAM, the way it was when 16MB was considered a lot.

And looking quickly at the Intel 925X MCH (memory controller hub) registers, nothing jumps out as a good candidate for some obvious bug.

Linus

Here is the PCI controller chain to the device:

```
00:1c.0 PCI bridge: Intel Corporation 82801H (ICH8 Family) PCI Express Port 1 (rev 02) (prog-if 00
[Normal decode])
Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR-
FastB2B-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR-
<PERR-
Latency: 0, Cache Line Size: 32 bytes
Bus: primary=00, secondary=02, subordinate=02, sec-latency=0
I/O behind bridge: 00005000-00005fff
Memory behind bridge: fff00000-000fffff
Prefetchable memory behind bridge: 00000000fff00000-00000000000fffff
Secondary status: 66MHz- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- <SERR-
<PERR-
BridgeCtl: Parity- SERR- NoISA+ VGA- MAbort- >Reset- FastB2B-
Capabilities: [40] Express Root Port (Slot+) IRQ 0
Device: Supported: MaxPayload 128 bytes, PhantFunc 0, ExtTag-
Device: Latency L0s unlimited, L1 unlimited
Device: Errors: Correctable+ Non-Fatal+ Fatal+ Unsupported+
Device: RlxdOrd- ExtTag- PhantFunc- AuxPwr- NoSnoop-
Device: MaxPayload 128 bytes, MaxReadReq 128 bytes
Link: Supported Speed 2.5Gb/s, Width x1, ASPM L0s, Port 1
Link: Latency L0s <1us, L1 <4us
Link: ASPM Disabled RCB 64 bytes CommClk- ExtSynch-
Link: Speed 2.5Gb/s, Width x0
Slot: AtnBtn- PwrCtrl- MRL- AtnInd- PwrInd- HotPlug+ Surprise+
Slot: Number 16, PowerLimit 10.000000
Slot: Enabled AtnBtn- PwrFlt- MRL- PresDet- CmdCplt- HPIrq-
Slot: AttnInd Unknown, PwrInd Unknown, Power-
Root: Correctable- Non-Fatal- Fatal- PME-
Capabilities: [80] Message Signalled Interrupts: Mask- 64bit- Queue=0/0 Enable+
Address: fee0300c Data: 4169
Capabilities: [90] Subsystem: Giga-byte Technology Unknown device 5001
Capabilities: [a0] Power Management version 2
Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold+)
Status: D0 PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [100] Virtual Channel
```

Capabilities: [180] Unknown (5)

00:1c.4 PCI bridge: Intel Corporation 82801H (ICH8 Family) PCI Express Port 5 (rev 02) (prog-if 00 [Normal decode])

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-

Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR-

Latency: 0, Cache Line Size: 32 bytes

Bus: primary=00, secondary=05, subordinate=05, sec-latency=0

I/O behind bridge: 0000a000-0000afff

Memory behind bridge: f8000000-f9ffffff

Prefetchable memory behind bridge: 0000000080100000-00000000801ffffff

Secondary status: 66MHz- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- <SERR- <PERR-

BridgeCtl: Parity- SERR- NoISA+ VGA- MAbort- >Reset- FastB2B-

Capabilities: [40] Express Root Port (Slot+) IRQ 0

Device: Supported: MaxPayload 128 bytes, PhantFunc 0, ExtTag-

Device: Latency L0s unlimited, L1 unlimited

Device: Errors: Correctable+ Non-Fatal+ Fatal+ Unsupported+

Device: RlxdOrd- ExtTag- PhantFunc- AuxPwr- NoSnoop-

Device: MaxPayload 128 bytes, MaxReadReq 128 bytes

Link: Supported Speed 2.5Gb/s, Width x1, ASPM L0s, Port 5

Link: Latency L0s <256ns, L1 <4us

Link: ASPM Disabled RCB 64 bytes CommClk+ ExtSynch-

Link: Speed 2.5Gb/s, Width x1

Slot: AtnBtn- PwrCtrl- MRL- AtnInd- PwrInd- HotPlug+ Surprise+

Slot: Number 20, PowerLimit 10.000000

Slot: Enabled AtnBtn- PwrFlt- MRL- PresDet- CmdCplt- HPIrq-

Slot: AttnInd Unknown, PwrInd Unknown, Power-

Root: Correctable- Non-Fatal- Fatal- PME-

Capabilities: [80] Message Signalled Interrupts: Mask- 64bit- Queue=0/0 Enable+

Address: fee0300c Data: 4181

Capabilities: [90] Subsystem: Giga-byte Technology Unknown device 5001

Capabilities: [a0] Power Management version 2

Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold+)

Status: D0 PME-Enable- DSel=0 DScale=0 PME-

Capabilities: [100] Virtual Channel

Capabilities: [180] Unknown (5)

05:00.0 Ethernet controller: Marvell Technology Group Ltd. 88E8056 PCI-E Gigabit Ethernet Controller (rev 14)

Subsystem: Giga-byte Technology Unknown device e000

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-

Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR-

Latency: 0, Cache Line Size: 32 bytes

Interrupt: pin A routed to IRQ 14

Region 0: Memory at f9000000 (64-bit, non-prefetchable) [size=16K]

Region 2: I/O ports at a000 [size=256]

Re: Linux 2.6.22-rc2

[virtual] Expansion ROM at 80100000 [disabled] [size=128K]
Capabilities: [48] Power Management version 3
Flags: PMEClk- DSI- D1+ D2+ AuxCurrent=0mA PME(D0+,D1+,D2+,D3hot+,D3cold+)
Status: D0 PME-Enable- DSel=0 DScale=1 PME-
Capabilities: [50] Vital Product Data
Capabilities: [5c] Message Signalled Interrupts: Mask- 64bit+ Queue=0/0 Enable-
Address: 0000000000000000 Data: 0000
Capabilities: [e0] Express Legacy Endpoint IRQ 0
Device: Supported: MaxPayload 128 bytes, PhantFunc 0, ExtTag-
Device: Latency L0s unlimited, L1 unlimited
Device: AtnBtn- AtnInd- PwrInd-
Device: Errors: Correctable- Non-Fatal- Fatal- Unsupported-
Device: RlxdOrd- ExtTag- PhantFunc- AuxPwr- NoSnoop-
Device: MaxPayload 128 bytes, MaxReadReq 512 bytes
Link: Supported Speed 2.5Gb/s, Width x1, ASPM L0s L1, Port 0
Link: Latency L0s <256ns, L1 unlimited
Link: ASPM Disabled RCB 128 bytes CommClk- ExtSynch-
Link: Speed 2.5Gb/s, Width x1
Capabilities: [100] Advanced Error Reporting

Stephen Hemminger <shemminger@xxxxxxxxxxxxxxxxxxxxxxxx>

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>